August 1983 Volume 2 $2.00

softalk

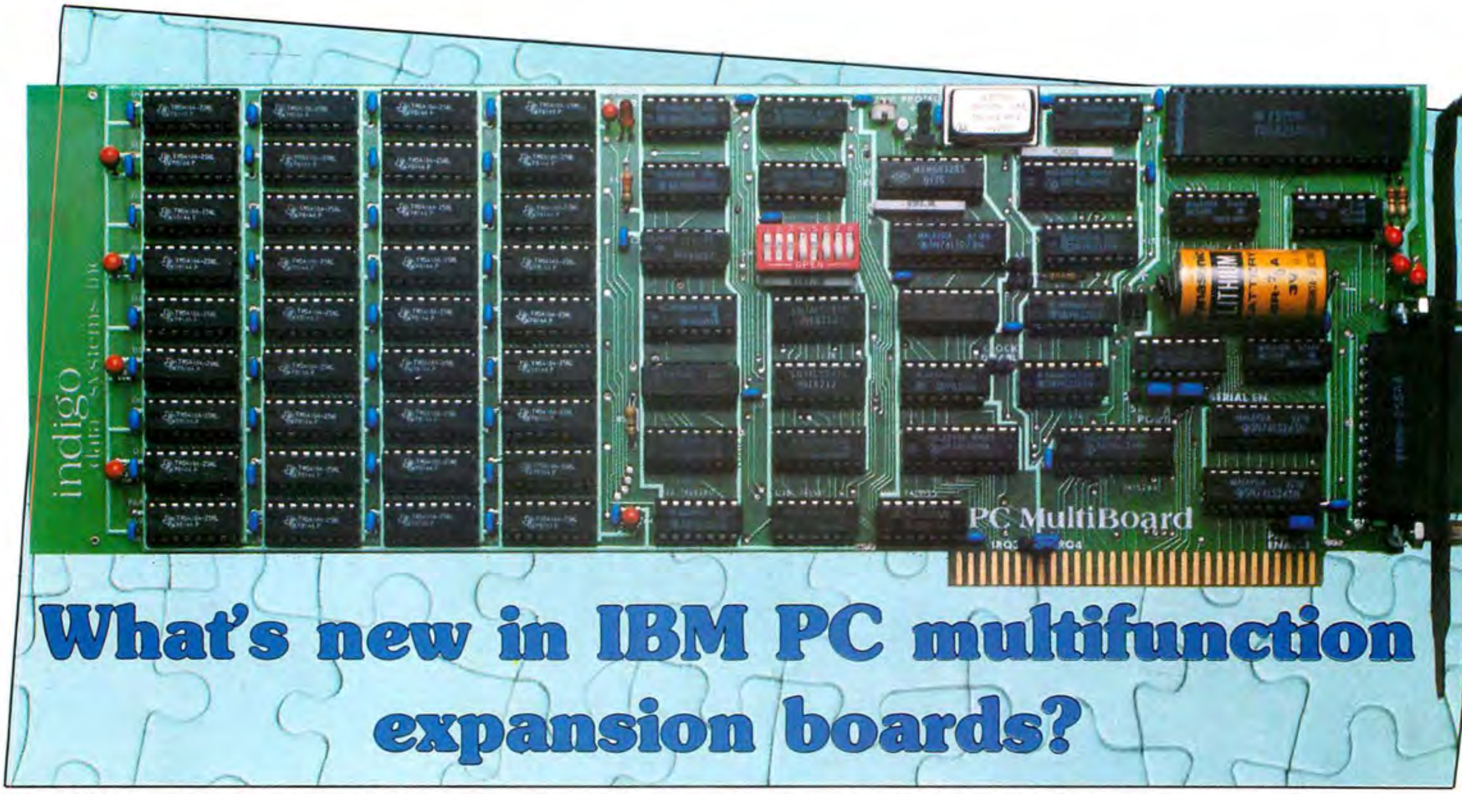
for the IBM Personal Computer

HOW BASIC STORES PROGRAMS

EXEC CONTINENTAL

FLYING THE OAG

A PC-POWERED SPACE SHUTTLE



Indigo has put together the most versatile expansion product available for the new IBM PC and IBM PC / XT

the PC MultiPak™

- **4 most requested hardware functions**
  The PC MultiPak™ includes a serial communications port, a real time clock calendar with a 10 year lithium battery, 0 to 256K memory and optional parallel printer port. The PC MultiPak™ without the parallel printer port is ideal for users with the IBM monochrome/parallel printer adapter installed.
- **Zero memory option**
  The PC MultiPak™ is available with no memory installed so you can fill your I / O needs today and still have memory expansion capability to an additional 256K for your future needs. This makes the PC MultiPak™ the perfect mate for the new IBM PC and IBM PC / XT.
- **4 software packages**
  The PC MultiPak™ includes our four most popular software packages. **Drive-It™** converts 32K to 320K of extra memory into an ultra-fast drive. **Spool-It™** utilizes up to 64K of extra memory as a printer buffer and eliminates valuable time spent waiting on the printer. **Print-It™** prints any IBM text or graphics screen to Epson printers. **Color-It™** prints screens to IDS printers. Color images can be printed on IDS prism printers. These software programs have received critical acclaim in Infoworld, Softalk, PC Age and Creative Computing magazines. Thousands have been sold separately. This $200 retail value is included in the PC MultiPak™ at no additional charge. DOS 2.0 Compatible.
- **5 year warranty**

$297
- ALL SOFTWARE INCLUDED
- NO INSTALLED MEMORY

64K-$365, 128K-$432, 192K-$499, 256K-$565
PARALLEL PORT OPTION ADD $30
64K MEMORY EXPANSION KIT $87
SOFTWARE AVAILABLE SEPARATELY
DRIVE-IT $59.95, **SPOOL-IT** $44.95
PRINT-IT $44.95, **COLOR-IT** $49.95

VISA

Indigo products may be found at **ComputerLand** and other fine stores.
Dealer And Customer Inquiries call TOLL FREE 1-800-231-9480 in Texas Call Collect (713) 488-8186

indigo
data systems, inc.

100 E. NASA Rd. 1, Suite 106, Webster, Texas 77598

softalk

for the IBM Personal Computer

Features

Departments

Index of Advertisers

Cover illustration by Craig Calsbeek.

Composition by Photographics, Hollywood, California. Printing by Volkmuth Printers, Saint Cloud, Minnesota.

IBM and Personal Computer are trademarks of International Business Machines, Armonk, New York.

 ISSN: 0733-2173. *Softalk for the IBM Personal Computer* is published monthly by Softalk Publishing Inc., 11160 McCormick Street, North Hollywood, CA 91601; telephone (213) 980-5074. Second-class postage pending at North Hollywood, California, and additional mailing offices.

Postmaster: Send address changes to Softalk/IBM, Box 60, North Hollywood, CA 91603.

Subscriptions: Complimentary trial subscription to all owners of IBM Personal Computers in the USA. If you own a pc but you aren't receiving *Softalk for the IBM Personal Computer*, send your serial number and mailing address to Softalk/IBM Circulation, Box 60, North Hollywood, CA 91603. Non-pc-owner subscriptions: $24 per year. Back issues (from June 1982): $3. *Softalk for the IBM Personal Computer* is totally independent of International Business Machines.

Problems? If you haven't received your *Softalk* by the fifteenth of the month, or if you have other problems with your subscriptions, Hal Schick can help out. Call (213) 980-5074.

Moving? Send new address and old to Softalk/IBM Circulation, Box 60, North Hollywood, CA 91603; telephone (213) 980-5074.

KNOW ANY GOOD ONE-LINERS?

The contest call this month goes out to the person who can create the best one-line program. It must be written in interpretive Basic, either disk or Advanced, graphic screen or mono. Write something nifty and original and be sure it runs. The most interesting program gets the big "hundred dollars' worth of software" prize.

Send your entry (or entries to One-Liner, Softalk IBM, Box 60, North Hollywood, CA 91603. Be sure to have the following information attached:

Name ______________________

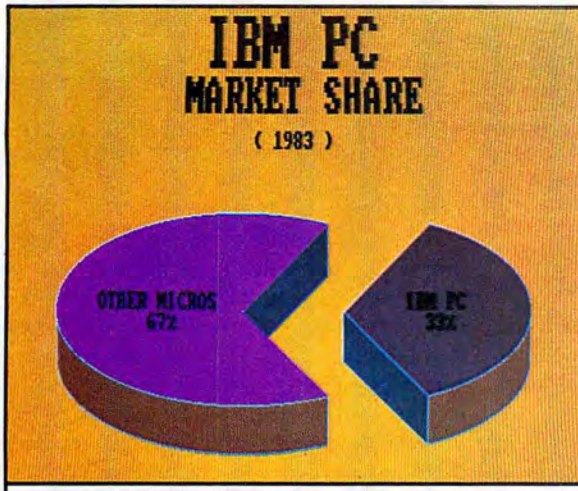
Address ______________ State ____ Zip ______

Phone ______________________

Name of Dealer ______________________

What I want if I win ______________________

(Prizes must be selected from advertisers appearing in this issue.)

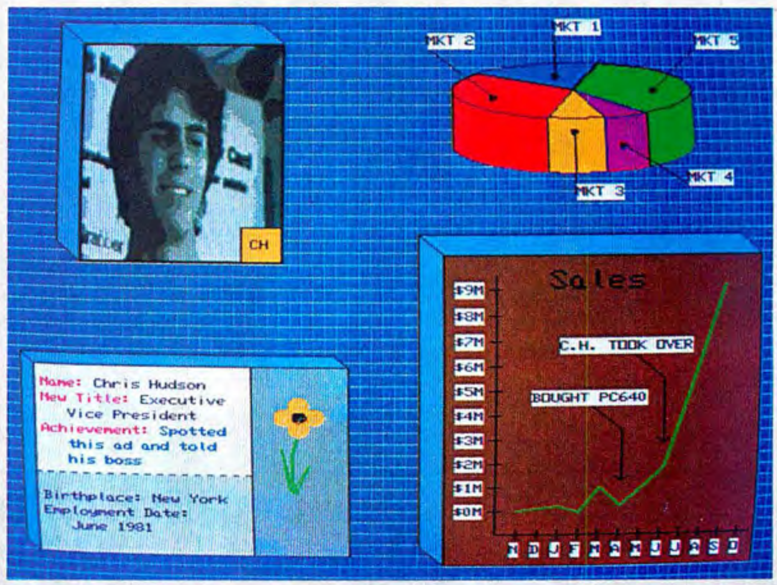


At Last!

Now there's a

640x480 Professional Color Graphics Board for the IBM PC...

Unretouched photos shot from 13" screen with Canon AE-1 SLR at f5.6, 1 sec. using Kodachrome 25 film



with

Dynamite Software!

Introducing

The PC640 Professional Color™ System

640 x 480, 16 out of 4096 color hardware, the HALO* software library of 58 graphics primitives, and the Cheetah* Fast Track end-user business graphics software system

from **SCION** Corporation
Your leader in color graphics
12310 Pinecrest Road (703) 476-6100
Reston, Virginia 22091 TWX: 710-833-0684

THE GOODS

Package 1 — $1595
The PC640 board, the HALO* software, User's Manual, video cable

Package 2 — $2495
Package 1 + Cheetah* Fast Track (turns your PC into a full business graphics workstation)

(High-res color CRTs also available)

* HALO is a trademark of Media Cybernetics, Inc.; Cheetah is a registered trademark of CENTEC Corp.

crosstalk

Making Book on It

I have found the early issues of *Softalk* to be much easier to read and therefore more informative than IBM's technical publications or any of the tutorials or "how-to" books I've run across. Do you contemplate collecting Beginners' Corner or System Notebook columns for commercial publication? They should make excellent guidebooks for new pc owners and I find that my back issues are eagerly copied by many of the new members of our UH PC Users Group.

If you are not planning publication, may our group obtain permission to photocopy small quantities for distribution among our members?

Frank S. Covey, Honolulu, HI

System Notebook and Beginners' Corner should both be available in book form from Softalk Publishing by the end of the year. In the meantime, please feel free to photocopy the columns.

Ode to Joystick

Much has been written about hardware and software support—or the lack of it—with respect to local dealers versus mail-order houses.

About six months ago I purchased an ABM Omni Board with a serial port, clock, printer port, and game adapter for my IBM pc. The game adapter was advertised to work with up to two Apple joysticks.

Recently I finally purchased one Apple-compatible joystick and hoped to play some games on my pc.

Much to my dismay, the joystick buttons didn't work properly with software I had purchased. After only one phone call and one letter ABM fixed the bug by sending me a terminator plug for the "phantom" second joystick. It understood the problem and solved it quickly and at no cost to me.

Service and support is not a matter of local dealers versus mail-order houses but is rather a matter of finding good people with expertise who care enough to help.

Ken Topolinski, Arlington Heights, IL

Zork Invades DOS—Film at Eleven

I recently discovered a DOS directory file on one of my disks that was composed mainly of the strange ASCII characters. Moreover, when I tried to load several of the DOS files that I knew were on the disk, I found to my surprise that they were no longer there. I then turned to the always useful Norton utilities. Norton's Disklook utility reported that there were no files on the disk. With increasing anxiety, I tried Norton's Secmod and began to examine the strangest collection of ASCII characters that I've ever seen on my disk files. Finally, after moving through several sectors of the disk I spotted a "hit troll with sword" and "pick up lantern." Heavens to *Zork*, what were these mutterings doing on a DOS disk? Further searching revealed that there were many other fragments of *Zork*-talk throughout the disk.

I then read the *Zork* manual carefully and indeed found that the *Zork* Reference Card for the IBM pc recommends a blank formatted DOS disk for saving game positions. Other readers might be interested to know that *Zork* is serious about this: It absolutely ruins a DOS disk with other files on it. I saved other *Zork* game positions on backup DOS disks and was able to reproduce the results described above. None of the Norton utilities seem able to reconstruct the original files on the *Zork*-invaded disk. Perhaps some other reader knows how to repair a DOS disk after *Zork* engages in battle with the existing files. *Zork* is fun, but it obviously can cause problems if it gets out of its underground empire.

Linda Baluchi, Ithaca, NY

Analyzing a Profitable Plot

I have enjoyed reading the Profit Plot in *Softalk*. I am interested in seeing articles on real-estate-investment analysis. I have done some of this myself and it works out really well using *VisiCalc*. A problem I have encountered is variable rate mortgages and I would like to see some methods of calculation for these (for example, where the interest rate goes up each year for a certain period of time and the unpaid interest accrues as part of the balance). With this type of loan, you can pay on it for five years and end up owing more than when you started. The spreadsheet format lends itself well to analyzing cash flow, depreciation, and estimated profit at time of sale. The internal rate of return would also be a useful calculation for evaluating properties.

Tim Hitson, Boulder, CO


DOS
Printer
in A
* DOS likes to announce itself by typing A>. The cursor sits next to the >. This is called the "DOS PROMPT". When you see the DOS prompt you can count on this:
DOS IS ALIVE AND IS PATIENTLY WAITING FOR YOU TO COMMAND IT.
GIVE DOS YOUR FIRST NAME

DOS 2.0: One Man's Story

Having just purchased and gotten my feet wet with DOS 2.0, I thought I might pass along some of the things I have learned (mostly the hard way).

First let me note that the package is well worth the price. There is at least one aspect that directly pays for itself and others that do essentially the same. The aspect that directly pays for itself, so to speak, is the example of a 180K device driver given in Chapter 14, "Installable Device Drivers." This is source code, in assembler, for a 180K virtual disk. It works precisely as it is given and uses up 672 bytes in a .com file; of course it requires the DOS 2.0 system for its use but all that it requires is to include the file Config.sys with the statement Device=filename in it (where filename is the name you have given the .com file for the virtual disk). It is easy to change the source code to create a 360K virtual disk by the following changes:

Description	Change	Line Numbers
sectors/allocation unit	2 instead of 1	88 and 111
number of directory entries	112 instead of 64	91 and 114
total number of sectors	720 instead of 360	92 and 115
Media descriptor	0FDH instead of 0FCH	93 and 116

The only other change is line 248 of the listing where you change the number of paragraphs to add from 2D00H to 5A00H (just twice). The only caveat is that the program must be precisely copied. The first time I did it, and after correcting various typing errors, I found it still would not run. The problem was (and believe me, I don't yet understand why) that on line 108 I had left only one space between IBM and 2.0, rather than the two spaces they have. It will run correctly if there are more than two spaces between IBM and 2.0, but not if there is less than two. I really do not understand that.

The next comment has to do with Chapter 13, "Using Extended Screen and Keyboard Control." The chapter is essentially unintelligible with respect to the implementation of the control sequences given. My travails with Big Blue on obtaining clarification would fill a book. I finally got enough of a clue (but not correct information) to lead me to a solution after some experimentation. It is as follows, and there are two methods, at least. One way is via Basic and the other is useful if you have some means of entering control characters (particularly escape) literally.

Create a file (in the root directory) that contains the various control sequences. Once that exists, and the statement *Device = Ansi.sys* is in your Config.sys file, the sequence will be automatically loaded on boot (not otherwise). The problem is the escape control code that I have been unable to find any way to enter from the pc in a direct way. However, I offer two other ways.

(1) Using Basic

Use the following program to create a sequential file containing the control sequences.

```
10 OPEN "filename.bat" FOR
   APPEND AS #1 'Opens
   filename.bat for sequential input
20 PRINT #1,CHR$(27);"[rest of
   commands after "[" "
30 CLOSE
```

The crucial thing is that this enters escape for you. If there is a string inside the " " , then just enclose that in single quotes (for example, *chr$(27,"[......'string'......").* What you can also do is to create a file containing just the *CHR$(27)* in line 20 and then use Edlin to add the rest on. That is, use it as a "seed" file. You can, of course, include more than one control sequence in a given file (separate lines, of course).

(2) Other Method

I happen to have a full screen editor (*VEdit*) that allows you to enter control characters literally. I use this to create the sequences. However, *VEdit* is a little tricky with 2.0. The minimum file size it creates is 128 bytes (this is an end of the file problem that the *VEdit* people tell me they are fixing in a revision for 2.0). What I do is call the file with Edlin and just close it, to get to a reasonable file size. Again, you could use the *seed* aspect after getting the first escape entered.

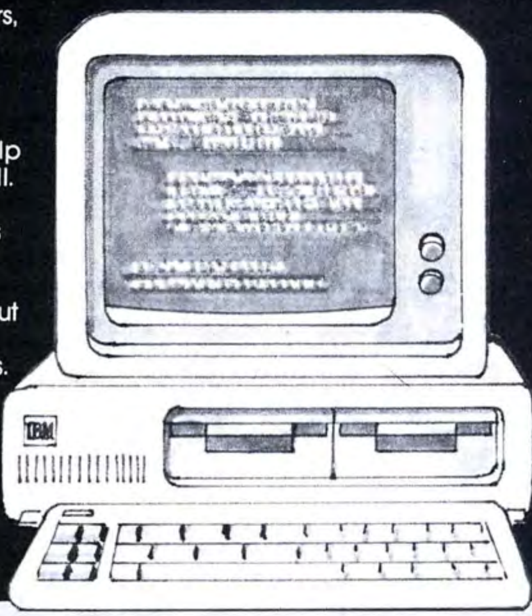
Now for the surprise. I was merrily going along thinking, boy, this is a nice cheap substitute for *Prokey* or *Keynote* or what have you. I started to create a series of macros to call with the key strokes, and then when I went to enter them (from an Autoexec.bat file) the system hung up on me. Some experimenting yielded the result that you can only use a total of about 300 bytes of files for Chapter 13 control sequences to stay away from the hang-up problem. I checked with Big Blue and they allowed as how, yep that's right, more than 300 bytes overlays some of Command com. Wouldn't it have been nice if they had told you that?

Consequently, if your requirements for keyboard sequences are limited to a total of

On the Fine Art of Family Business Management.

Keeping family business matters, under control is difficult ... Write this check ... Pay that bill ... Insurance premiums due ... All the paperwork adds up to a headache. You need all the help you can get to stay on top of it all. The **HOUSEHOLD AIDS** software package from N.F. Systems puts an end to guesswork, paper shuffling and haphazard accounting. Buy it today and put the power of the P.C. to work managing your family business.

N.F. Systems Software
for the
IBM PC



HOUSEHOLD AIDS

A group of six programs designed to organize and control paperwork for the entire family. **$49.95**

1 **HOUSEHOLD INVENTORY**—26 categories of possessions. Totals by category including cost and current value. Display/Print. Insurance companies love it!

2 **VITAL RECORDS**—keep lists of social security numbers, bank accounts, wills, credit cards, insurance, etc. A must for everyone!

3 **VCR INDEX**—also could be used to keep track of regular cassettes, or books. Unlimited tape numbers.

4 **RESUME**—keep it handy—keep it current. Keep resumes for several family members

5 **PERSONAL STATEMENT**—keep available to print as needed. Covers all topics. Works for any bank.

6 **NAME/ADDRESS/PHONE**—also, allows space for children's names and a miscellaneous field for birthdays. Christmas cards, etc. Search on last name.

STOCK PORTFOLIO

Major Features:

- **Enter Security Purchases/Sales**
- **Value Portfolio**
- **List Portfolio**
- **List Portfolio Transactions**
- **Create Year-End Tax Information**

This user friendly program allows you to keep multiple portfolios for you and your family. Use the power of the computer to track your investments. All reports can be displayed on the screen and/or printed on your printer. **$39.95**

ICARUS PROBE

Your mission, should you decide to take it, is as follows: Fly and land the Daedalus Space Probe to the inner-most chamber of the Panamon Complex on the planet Armitron. This will take you through many twisting tunnels which you must navigate in order to succeed. Failure means instant death. Your fuel supply is limited as you thrust through the mountainous terrain headed for the landing pads. An exciting game utilizing the monochrome display. **$34.95**

All use 1 disk and 40 or 80 character display. 64K except where noted.

MASTERCARD/VISA
Add $2.50 Postage & Handling

CATALOG AVAILABLE

N.F. SYSTEMS, LTD.
P.O. BOX 76363, ATLANTA, GA 30358

404-252-3302—VOICE 404-252-4146—DATA

300 bytes, this will save you some money, otherwise no. Some other observations. The tree-structured files are really great. Using them, and the *sort, find* and redirection, I have been able to create a disk filing system that is quick and easy to use and update. DOS 2.0 also includes a print spooler that works quite nicely (but the files must be in the current directory to use it) as well as other goodies. The only thing DOS 2.0 lacks is a decent uniform exposition. The manual is very spotty. Some parts are written quite clearly, but for others obtuse is an insufficient description, the champion obtuse being Chapter 13 (if it had been Chapter 11, the match would have been perfect).

A final word, for completeness' sake, about the virtual disks. I have an ordinary pc with two disk drives. The virtual disk comes up as *Drive D>*. After some playing around, it appears that the DOS 2.0 system allows for one fixed disk (hard disk) after the number of regular disk drives logged on with the switch settings. If you had three disk drives, then the virtual disk would call up as *E>*, and so on.

A word about *EasyWriter*. I purchased *EasyWriter II* shortly after it came out and found quite readily that it utilized single-side formatting for the crucial aspect of file transfers between DOS and *EW II*, as well as for the size of its disks. I wrote IBM a letter saying that its manual was misleading (to say the least) and almost immediately got a phone call saying that it was doing an update to correct that and that it would be out in a few months (the few months turned out to be six). IBM has been less responsive with respect to other letters about other aspects of *EW II*.

Before closing, I would add that if any readers want a run-time copy (the com files) of the 180K and 360K virtual disks, I will be happy to send them a copy if they send me a blank disk, return mailer, and return postage, or alternatively $5 to cover costs.

Morton F. Kaplon, Pomona, NY

The HP Handshake

In response to the letter "PC Calling HP" in the June issue of *Softalk*: Tymlabs Corporation has developed a terminal emulator package that allows the IBM Personal Computer to emulate a Hewlett-Packard 2621 terminal.

This package recognizes HP control characters and utilizes the ENQ/ACK handshaking protocol to communicate with the HP 3000 computer.

With this emulator you can use the IBM Personal Computer to run any character mode application. If you have programs that use HP's block mode screen handler VPLUS/3000 (VIEW), Tymlabs markets a product that runs on the HP 3000 called PreView that allows programs using the block mode VPLUS/3000 to run on any character mode CRT.

Teresa L. Norman
Tymlabs Corporation
Austin, TX

In the June 1983 "Crosstalk," Don Leifheit asked about the availability of software to allow the IBM pc to interface with Hewlett-Packard 3000 Series computers. Our company, Inner Loop Software, has a new product called VDTE 2 that emulates an HP2624B terminal. It is especially designed to communicate with HP3000 computers, both as a terminal and for file transfer.

Richard Gillmann
Inner Loop Software
Los Angeles, CA

A Hard Disk Is Gonna Fall

Please print this before another hard disk is inadvertently erased. IBM's DOS 2.0 fails to give any precautions when using the format command. It's possible for a weary user to type format without specifying drive A, and, in doing so, erase the entire hard disk.

The following batch file will prevent this catastrophe. Since it's unlikely that any user would ever want to format any disk besides the one that is in the floppy drive, this procedure changes the command so that all formatting routines occur on drive A only.

First, rename Format.com to Doformat.com. Then create a batch file called Format, which will cause the disk in drive A to be formatted. You may also include the /s parameter in the doformat command to put the system files on the disk. Here's the procedure:

```
A>RENAME FORMAT.COM DOFORMAT.COM
A>COPY CON:FORMAT.BAT
DOFORMAT A:
(press key F6 and return to writhe the file to disk.)
```

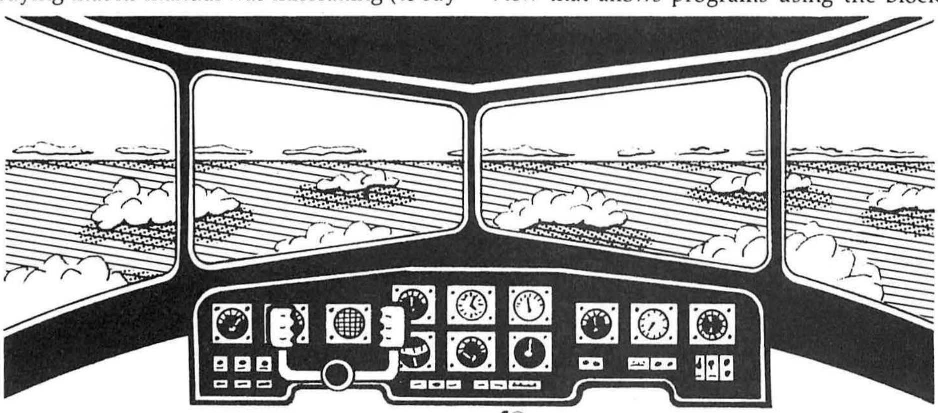
To utilize the new procedure, type format to format drive A. The batch file will do the proper job from there on.

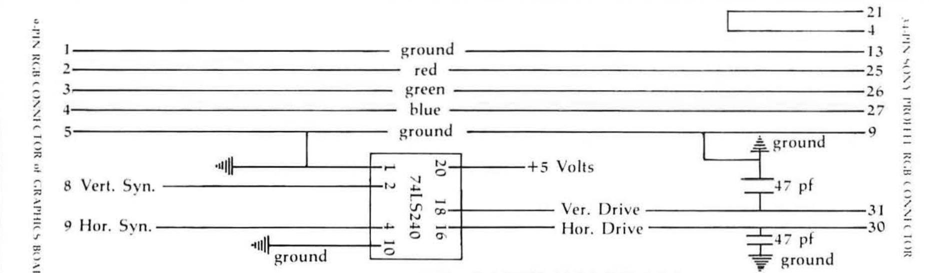
Tom Sheldon, Santa Barbara, CA

pc/Profeel Inverter

When I looked at the features of Sony Profeel Monitors, I was impressed and decided to get one for my IBM pc. The Profeel Monitor has a provision for RGB input and composite video and also has a stereo amplifier. The monitor can be used as a high-quality TV with the optional tuner. I could buy a nineteen-inch Profeel Monitor at a discount store for $568. The resolution of the monitor matches well with the color graphics of the IBM pc and supports an eighty-column display.

When I connected the cable between the pc RGB output and the monitor, to my surprise I found that the screen kept scrolling diagonally. The monitor did not correctly in-





INVERTER AND CABLE DIAGRAM

terpret the horizontal and vertical synchronization signals from the pc. The IBM product center was not of much help in solving this problem. With the help of a friend, Dan Nosenchuck at Caltech, I made an interface to use the Sony Profeel with the IBM pc. Many of your readers might be interested in this interface.

The problem here is that Sony employs negative TTL logic and the IBM pc color/graphics RGB output uses positive TTL logic. An interface should therefore consist of an inverter to change the polarity of the horizontal and vertical synchronization signals. The simple circuit and cable for interfacing, as shown in the diagram, can be made in less than an hour and for under fifteen dollars. The five volts supply to the inverter can be obtained from Pin 5 of the light pen connector or Pin 4 of the speaker interface without any soldering on the board, using a simple connector that can be obtained from any electronics supply store.

Dr. Malladi Subbaiah, Palo Alto, CA

Physician Software

This letter is in reference to Dr. F. A. Choudhry's query in *Softalk* (January 1983) regarding software for medical practice. We have developed a Medical Office System software package that runs on the IBM pc. It handles all aspects of patient and family billing, insurance-claims processing, daily and monthly analysis of transactions, accounts receivables, appointments and medication scheduling, and many other things. We suggest Dr. Choudhry contact Innovation Systems International, Box 444, Hopewell Junction, NY 12533, (914) 226-8730.

Ram Dixit
Innovation Systems International
Hopewell Junction, NY

Reply to the Clergy's Query

In reply to the clergyman ("Questions & Answers," June 1983) who was looking for mass storage systems for his pc, the 512K JRAM board from Tall Tree Systems will work with a 2.5 MB electronic diskette. Also, diskette storage of 2.5 MB is possible using a Drivetec Drive and JFORMAT software, 1.25 MB per diskette using a Mitsubishi drive and JFORMAT, and over 800K bytes per diskette with an 80-track 96-tpi drive and JFORMAT.

Of course one does leave the vanilla blue IBM environment somewhat with these methods, but then it seems to me that finding solutions on top of what IBM salesmen provide is one of the reasons for question-and-answer sections in magazines.

Martine H. Boot
Tall Tree Systems
Los Altos, CA

A Financial Masterstroke

The review of personal and home finance programs for the IBM pc in the June issue of *Softalk for the IBM Personal Computer* failed to mention one of the friendliest and fastest of the personal financial programs, namely *Personal Finance Master* by Spectrum Software.

Personal Finance Master sells for only seventy-five dollars and easily outdistances most of the programs you reviewed in your article when it comes to speed and simplicity while still retaining all the features of the other programs. In addition, it keeps track of addresses and prints them automatically on the checks. This is a real timesaver when paying the monthly bills. Editing transactions is fast and easy, and there is no faster program for reconciling a bank statement. For a real eye-opener, try reconciling a bank statement with *PFM*. Programs like *Home Accountant Plus* suddenly seem cumbersome and antiquated in comparison. A real bonus is that what you see on-screen or in a printout looks like a real checkbook that we are all familiar with.

As far as I am concerned, you missed the best or at least one of the best programs available for home or small business accounting.

Check out *Personal Finance Master*. You might just be impressed!

Charles A. Coakwell III, M.D., Mayfield Heights, OH

Strung Out on WordStar

I am, with many others, a *WordStar* addict. But, like most addicts, I require higher quality with increased use. I have but two major gripes with *WordStar*, and am hoping you can direct me to remedies. (1) How can I disable the paranoid function in *WordStar* that creates a backup copy of a file? I am perfectly capable of making my own backups before I alter a document and don't want to

be saddled with using up unnecessary disk space or with deleting the .bak file before opening a document file. (2) Why is it that *WordStar* takes so much time to move through a document? I have 256K, so that's not the problem. I have run *Volkswriter* a bit, and though I don't think it is as good as *WordStar*, it does have the advantage of immediate run-through. Can this deficiency be rectified?

Lee Clarke, Stony Brook, NY

Since Life Itself?

In reply to Mr. Cotten ("Crosstalk," June 1983), the pc can be a "tool to be used within the business community as a problem solver" but it is also much more. I consider it one of the most versatile creations since life itself. Although I use the pc for Cobol eight hours a day, when I get home at night it sings, plays games, and even asks my girl friend to cook dinner. "The boss is coming key" from FriendlySoft is yet another innovative idea for the pc.

David Stinson, Memphis, TN ▲

WINNERS

In the *User's Group Scramble* contest in the May issue, the oddly monikered group didn't discuss "Hepsoedu kortl esalof wrpe optean." In fact, they barely discussed anything because "Software spoke louder than people" at their meeting. The first phrase is made up of all the letters, in order, that were then to be rescrambled to form the winning phrase.

It was a feat that not all entrants were able to perform successfully. Of those few who did, the random-number generator picked Robert M. Gillis (Birmingham, AL) as the big $100 winner.

But take heart, tearful losers, as well as anyone else reading this. Through a typographical quirk of fate, the deadline for choosing the winner from the five limericks printed in this column last issue was erroneously listed as July 15. We're extending it to August 30, and we'll also give, to the first voting card chosen randomly, fifty dollars' worth of software.

So pull out your June issues and turn to page 16; check out paragraph four of *Contest Winners* and do what it says. There may be money in it for you. One more thing: Be sure to tell us what you want if you win. Prizes must be chosen from the wares of advertisers appearing in this issue.

Next issue we'll confess to a couple of tiny bugs in the whopping three-level June Jumble contest and show you how clever winners can be. ▲

ANNOUNCING THE END OF WORD PROCESSOR CONFUSION

It's one confusing business, choosing a word processor. All those companies assuring you theirs is the most fantastic one of all. And leaving you to cut through all the smoke and hopefully whittle them down to the best one for you.

Well, we can help. If you'll just invest the next three minutes reading this, we'll tell you about a word processor that makes picking the right one a real breeze.

FIRST, A WARNING. There are two things you must do when evaluating any word processor. First, be careful. It's no secret that many of today's claims about being easy to learn and use just don't *stand* up to careful comparison. Watch carefully for complicated codes and programs that require "training sessions." Secondly, be selective. Buy a word processor you can use on a daily basis, not one that requires another "training session" when someone goes on vacation. In other words, be very careful to select the word processor that's absolutely the best and easiest to use. Which is what we'll now introduce you to.

WHY ONE WORD PROCESSOR MAKES THE MOST SENSE OF THEM ALL. Meet WordPlus-PC. It's the advanced, "user-oriented" software package that turns your IBM-PC computer into a word processor equipped with the sophisticated capabilities of far larger, more expensive systems.

THE REMARKABLE WORDPLUS-PC STORY. In a nutshell, it's this. WordPlus-PC was designed from the ground up to be the very latest in word processing technology. It has all the important features you could ask for. And the program is so easy to learn and use it defies comparison with other software.

HOW WE CAN BE SO SURE OF OURSELVES. We're Professional Software, a leader in word processing software with over 30,000 customers contentedly churning out letters and documents around the world on personal and business computer systems. We know precisely what a top quality word processor must do. And we've studied all the shortcomings of our competitors. That's how WordPlus-PC came to be the easiest-to-use word processor ever.

ALL THE FEATURES YOU'D EXPECT.

WordPlus-PC is both powerful and versatile. Its long list of major functions includes movement of paragraphs, global search-and-replace, and creation of multiple "boilerplate" and personal letters with the touch of a few buttons. There are 3-line headers and footers and automatic page numbering. When you underline or boldface words as you type, WordPlus-PC displays them in that form on the screen. You can merge information created by popular spread sheet programs. You'll create personal letters quickly and easily, or merge name and address information from other programs. You can easily create standard layouts for preprinted forms and use WordPlus-PC to prepare invoices or quotations. For large corporate customers, WordPlus-PC has the capability to merge information from files created by the firm's mini or mainframe computer. You can even type new documents while the system is printing other ones. And you'll find WordPlus fully compatible with virtually all popular ASCII letter-quality and dot matrix printers.

A BREEZE TO LEARN AND USE.

With WordPlus-PC's easy-to-use documentation, you'll be using your word processor right from the first minute. After an hour, you'll feel at home with all the basic functions and ready to explore the more sophisticated ones. After a few hours you'll put the manual away. There are never any complicated or confusing codes to memorize or look up. Rather, the screen's "message area" reminds you how to do what you want. WordPlus-PC's unique keyboard color coding makes things even easier. And whenever there's a question, simply press the HELP function to get back on track.

ALL THE HELP YOU NEED VIA TOLL-FREE PHONE. Learning something new goes best when you have encouragement, reassurance and answers to questions. That's how WordPlus-PC helps you learn. In addition to the help you'll get from your dealer, we're ready to assist you when you call our special toll-free number and talk to one of our carefully trained advisors. This service is available at no charge during the 90 days following receipt of your user registration card. It's all part of Professional Software's total commitment to customer service and support.

SEE WORDPLUS-PC FOR YOURSELF AT YOUR COMPUTER DEALER. Call us today. We'll answer all your questions. And we'll give you the name of the Professional Software dealership nearest you. Then go witness WordPlus-PC in action. Because once you're sitting in front of WordPlus-PC you'll quickly see where all our confidence and enthusiasm come from. After all, you already own the best personal computer. The best word processor is only $395. So call us today.

CALL US TOLL FREE:

1-800-343-4074

Professional Software Inc.

51 Fremont Street, Needham, MA 02194 Telephone (617) 444-5224

Dealer, distributor, OEM manufacturer, and corporate inquiries are invited. Call us at (617) 444-5224.
WordPlus-PC is a trademark of Professional Software Inc.
Specifications are subject to change without notice.
WordPlus-PC was designed and written by Andres Escallon.

SOFTALK CLASSIFIED ADVERTISING

Softalk/IBM's classified advertising section offers a considerably less expensive way than display advertising to reach tens of thousands of IBM Personal Computer owners.

Classified advertising space is available at the rate of $10 per line for the first ten lines, with a five-line minimum. Each line over ten lines is $25 per line. Ad copy should be received no later than the 10th of the second month prior to the cover date of the issue in which you want the ad to appear. Payment must accompany ad copy.

The publisher reserves the right to reject any advertising that he feels is not in keeping with the publication's standards.

Heads will be set in 10-point boldface, all capitals only. Italics are available for body text only; please underline the portions you would like italicized.

The body text of the ad will hold roughly 45 characters per line. Spaces between words are counted as one character. Heads will hold roughly 24 characters per line, with spaces between words counted as one character. Please indicate whether you would like the head centered or run into the text.

Please write or call for additional information.

Softalk/IBM Classified Advertising
11160 McCormick
Box 60
North Hollywood, California 91603
Attention: Linda McGuire
213-980-5074

There's only one word processor in the world that has a 30,000 word dictionary and sells for under $100.

And this is it.

introducing

FriendlyWriter™
with FriendlySpeller™

Word processing doesn't have to be a gruelling job.

The best little letter writer you've ever learned to use in 10 minutes is finally here. And it comes with something no other W.P. software in its price range has — a 30,000 word dictionary/spell checker.

FriendlyWriter is designed specifically for the kind of word processing you do the most — 1 to 6 page business and personal letters.

Using FriendlyWriter is as easy as striking a key.

All functions and commands, from setting your margins, to moving or inserting a block of text, can be executed with A SINGLE KEYSTROKE.

FriendlyWriter will help you produce letters you'll be proud to mail, and FriendlySpeller will make sure your letters read as well as they look.

SYSTEM FEATURES: 30,000 + WORD DICTIONARY * USER DEFINED LETTER FORMAT * AUTOMATIC L/R JUSTIFICATION * SINGLE KEYSTROKE COMMANDS * COMPLETE CURSOR CONTROL * UNDERLINING * MOVE, INSERT, DELETE LETTERS, WORDS, BLOCKS * LETTER RECALL BY NAME, DATE, CLASSIFICATION * MULTIPLE COPY PRINTING * TWO DISKETTE PKG. * REQUIRES: 64K, PC DOS, ONE DRIVE, ANY MONITOR.

FriendlySoft, Inc.

ARLINGTON, TEXAS

All FriendlySoft products are available through an international network of dealers including COMPUTERLAND STORES, IBM PRODUCT CENTERS, COMPU-SHOPS, and ENTRE COMPUTER CENTERS.

QUESTIONS & ANSWERS

by Nancy Andrews

Q: Is there any way to inhibit the "Redo from Start" error message that occurs when a Basic *input* statement encounters an incorrect data type or number of variables? I would like to branch to an error-handling subroutine and print a more descriptive error message, but there doesn't seem to be an error code available to accomplish this. Any help you could give would be greatly appreciated.

Larry R. Scott

A: You have indeed described an irritating feature of Basic on the IBM pc. The *on error goto* statement doesn't trap this kind of problem. Probably the best way to avoid "Redo from Start" is to remove the possibility of error at the *input* statement level. To do this, input your data into a string variable. Then do your own validity checking and convert the data into numeric values (if numbers are what you want). This way you can also write your own error-handling routines and messages.

Q: I would like to know how to obtain random numbers at the machine language level. I have read DOS, the *Macro Assembler*, and the *Technical Reference* manual, and I still haven't the slightest idea how to do this. I would like to write games in machine language, but I can't get off the ground without random patterns.

Andy McGuire

A: The bible for doing this kind of programming is Donald Knuth's *The Art of Computer Programming*. Volume 2 contains information on hash functions; you can use a hash function to generate random numbers. The typical hashing formula is:

(Seed * A + B) MOD C

The trick is to choose values for A, B, and C that have good properties. Good properties include numbers that are relatively prime and are not powers of 2. Knuth goes into great detail about good properties. Happy reading!

Q: I'm having a problem with the IBM *Asynch* package. It doesn't seem to work with a 132-column printer; it puts a carriage return in after every seventy-nine characters. Can you help me fix this?

Lee Beckley

A: The IBM *Asynch* package does not add carriage returns to the data it sends to the printer; it merely sends on streams of bits and does no interpretation. So the problem is elsewhere. If the data you are reading contains a carriage return every seventy-nine characters, the printer will, of course, faithfully print it, and you will need to write a little program to remove the unwanted returns. If the carriage returns are not in the data you are receiving, you may just need to use the *mode* command to set your printer to 132 columns before you print.

Q: In preparing commercial software, how can I incorporate a no-copy lock on the program file (a la *Microsoft Adventure*)?

Raymond S. Munn

A: Copy protection is a controversial issue, and many software developers are deciding against copy protection. Many protected programs cannot be used with RAM disks; hard-disk owners may want several copies of a program in different directories on their disks; and there are very few copy-protection schemes (probably none) that someone hasn't been able to break.

Now, after all that, if you still want to copy-protect, there are two basic schemes—changing sector size and changing the order in which sectors are read.

To change sector size you need to change disk base parameter 4, the fourth byte in the disk parameter table (see *Technical Reference* manual, page A-41). Parameter 4 is set to 2, making each sector 512 bytes. To make a sector 256 bytes, change this parameter to 1. You could read one 256-byte sector, then change the parameter back. This change would make it impossible to copy the disk by means of the *diskcopy* command.

To change the order in which the sectors are read, you need to change parameter 5, EOT. This is usually set to 8 (or 9 with DOS 2.0). To change its value, use interrupt 1E (see *Technical Reference* manual, page 3-21). A person could use *diskcopy* to copy this disk, but the program would not work correctly because the sectors would be read in the wrong order.

Q: I have a pc-XT and have two questions.

1. I can load a menu program from a batch file, but I can't load and run a batch file from a menu program. Is this possible? If so, how might it be done?

2. Is there any way to control the numlock and capslock features from a Basic program? It would be nice to have a program that

Teach yourself software in minutes. Not hours.

Insert an ATI training disk, and presto! Three minutes later, you're practicing usable skills. Within an hour, you're proficient.

Even if you've never touched a computer before.

ATI's self-prompting disk simulates the actual software on the screen of your personal computer. If you have a question later, the helpful handbook is set up to give you the answer in 15 seconds.

ATI Training Power is faster because it's Performance Based. Instead of taking time teaching you how a program works, ATI shows you how to put it to work. Right away.

ATI's unique Performance-Based Training is widely used by many Fortune 500 companies, such as IBM, DEC, Xerox, NEC, and Toshiba. Not to mention thousands of small businesses.

ATI's training programs operate in virtually every business oriented microcomputer.

Match ATI's selections with your software library, and order today.

Word Processing

ATI's Word Processing Series teaches you how to create, edit, move, and print text with popular programs, fast! A must for executives and secretaries alike.

Financial Planning

What if you could learn to do a complete "what if" spreadsheet analysis in less than an hour? ATI's Financial Planning Series teaches you this and more. Turn software packages into powerful management tools, quickly.

Database Management

To get the most out of your data base, learn to use these programs fast, with ATI's Database Management Series.

Operating Systems

Learn to prepare, monitor, maintain, and troubleshoot your own system like a pro, with ATI's Operating Systems Series.

Accounting

Now it's easy to put your accounting on computer, with an outstanding software package and ATI Training Power to get you (and your bookkeeper) up to speed, rapidly and cost-effectively.

Programming

ATI's Basic Programming Series teaches you how to write, enter, debug, and use programs that analyse business expenses, organize mailing lists, and more.

Take this ad to your dealer. Ask if you qualify for your FREE ATI SOFTWARE SAMPLER (IBM, Apple & 8" formats).

Please rush me ATI Training Power™ programs, at $75 each, for this software:

Integrated Software
- ☐ **Lotus 1-2-3 (NEW!) (Available August 15th)**

Database Management
- ☐ **dBASE II-vol 1&2**
- ☐ **EasyFiler**

Word Processing
- ☐ **WordStar-vol 1&2**
- ☐ **EasyWriter II**
- ☐ **Perfect Writer**
- ☐ **Spellbinder**
- ☐ **Benchmark**

Financial Planning
- ☐ **Multiplan**
- ☐ **SuperCalc**
- ☐ **VisiCalc ($50) Limited offer**
- ☐ **Perfect Calc**
- ☐ **EasyPlanner**
- ☐ **Microplan**

Operating Systems
- ☐ **PC DOS**
- ☐ **MS DOS**
- ☐ **CP/M**

Programming
- ☐ **BASIC**
- ☐ **APPLESOFT BASIC (Available August 15th)**

Accounting
- ☐ **BPI Gen. Acct.**
- ☐ **Peachtree General Ledger***
- ☐ **Peachtree Accounts Receivable***
- ☐ **Peachtree Accounts Payable***

*PC(MS) DOS formats only

Enclosed is $75 each plus $2.50 shipping. VisiCalc Trng $50 (CA add 6.5%)

Name ______________________ Phone ( ) ______________

Street ______________________________________

City ______________ State ______________ Zip ______________

My computer uses:
- ☐ **PC DOS** ☐ **CP/M-86** ☐ **XENIX**
- ☐ **MS DOS** ☐ **APPLE II CP/M**
- ☐ **CP/M** ☐ **APPLE IIe 80 col.**

My computer brand is ______________

Mail order form to:
Software Training Company
A Division of ATI
3770 Highland Avenue, Suite 201,
Manhattan Beach, CA 90266

Dealers: call for sales kit
(213) 546-4725

FOR RUSH ORDERS CALL: (213) 546-5579

VISA MASTERCARD

ATI Money Back Guarantee
If you're not completely satisfied with how fast you learn with ATI Interactive Training Programs, simply return it within 3 days for full purchase refund.

ATI Training Power™

would turn on or off the keypad numbers or capital letters.
John W. Foster

A: Assuming that your menu program is written in Basic, you can use the *shell* command in your program to execute a DOS command, such as a batch file. The format you'd use is:

```
SHELL "MYPROG.BAT"
```

You can turn capslock on and off from a Basic program. The simplest way to change from lower case to upper or from upper case to lower is to put these statements in your program:

```
DEF SEG = 0
POKE &H417, PEEK (&H417) XOR &H40
```

This will toggle capslock, turning it on if it's off or off if it's on.

You can also turn the number keypad off and on with similar statements:

```
DEF SEG = 0
POKE &H417, PEEK (&H417) XOR &H20
```

The only change is the value following the *xor*; the numlock bit is in location hex 20.

For a thorough examination of the memory locations where keyboard attributes are stored, see "The Scroll Lock Mystery Solved" and "Questions and Answers," both in *Softalk*, March 1983.

Q: I have *Volkswriter* and wish to set the capslock function to on when I boot. I could do this from BasicA if I were using BasicA. But I can't do this from *Volkswriter*. Is there a value into location 40:17?
Ed Smuckler

A: The Lifetree Software people say that with the current version of *Volkswriter* there is no way you can set capslock when you boot. However, their new version, *Volkswriter Deluxe*, will let you do this. It should be available by the end of the summer, and if you are a registered *VW* user you can get the scoop in their user's newsletter.

Q: Could you explain how the DOS 2.0 *graphics* command works? The manual says it can be used to print text mode screens, but when I do that I get the same output I get when I simply use the printscreen key without the *graphics* command. Also, I find that I don't get a good copy of the screen. For example, I made a double-line box (using ASCII characters 200, 201, and so on), but when I printed it, it came out a single-line box. Is the IBM printer not really compatible with the pc screen?

One last question. If I use a Basic program to create a screen, how am I supposed to return to DOS and issue a *graphics* command without messing up the screen with a lot of typing?
Michael Trombetta

A: The DOS 2.0 *graphics* command in text mode does exactly the same thing as hitting shift-printscreen. The reason you're not getting the double-line box is that the IBM printer does not have all the IBM graphics characters; as you've discovered, it doesn't have, among others, the double-line graphics characters.

The way to use the *graphics* command from within a Basic program is to use the Basic *shell* statement. The statement to put in your program is:

```
SHELL "GRAPHICS"
```

When you do this, the screen created by your Basic program should begin to print. ▲

When you bought a computer, you thought you could manage information easily.

Now you really can.

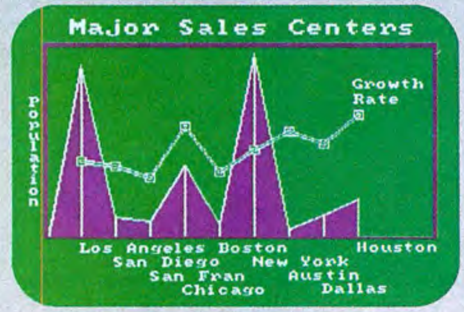
Introducing the most flexible manage

GETTING A GRIP ON INFORMATION:

Have you ever lost a contract or misplaced an important piece of correspondence? Have you ever dreaded the thought of working late just to put together a report for tomorrow morning's sales meeting?

Have you ever just felt that you can't get at the information that surrounds you? It can be very frustrating. Not to mention unproductive. But it doesn't have to be.

UltraFile has an easy-to-use graphing system: "Point" to the kind of graph you want to see (there are many to choose from). You can even do calculations right on the screen!



UltraFile can help you get a grip on all the information in your life. Whether you need to process large amounts of information in business or just want to catalog your home library.

UltraFile can do it for you. Because it's as powerful as it is easy to use.

UltraFile is the best reason there is for owning a computer.

SAVE COUNTLESS HOURS.

UltraFile can manage and organize your information in hundreds of different ways, no matter what your need. Here's just one example:

Suppose your company does business nationwide. You as sales manager know sales can be increased dramatically by concentrating on cities that have a population of at least 500,000 or an average income of at least $9,500. How can you find out which cities fit into those categories?

UltraFile can tell you. And do the calculations. And do the selections. And do the formatting. And give you the information that meets your criteria. Instantly.

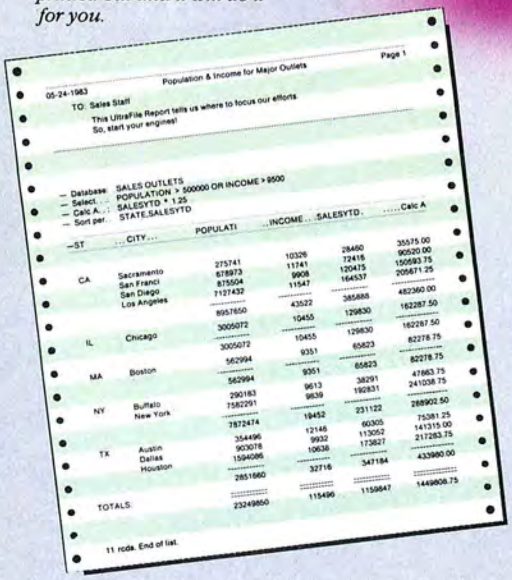
UltraFile can pull data out of a file of information and do calculations and even make projections.

If you just want to browse or look through your "file," you can choose to see every other item or even every seventh item, forward or backward. It's up to you. When you find the information you want, you "point" to it and it is automatically highlighted on your screen, keeping you from losing your place.

But more importantly, the information is automatically "remembered." And that means you can get the interesting or highlighted information sorted, calculated and formatted any way you want. Instantly. UltraFile even lets you add new columns of information and make revisions at any point without having to start all over. Unlike other programs, UltraFile is very forgiving if you change your mind.

Now suppose you'd like to have a printout of the information on your screen. Creating a "report" is easy. Tell UltraFile in plain English exactly how you'd like to see the information and it will do it for you. It's extremely flexible.

It's easy to create a report with UltraFile: Just tell UltraFile in plain English exactly how you'd like to see the information printed out and it will do it for you.



Population & Income for Major Outlets

Page 1

05-24-1983

TO: Sales Staff

This UltraFile Report tells us where to focus our efforts.
So, start your engines!

— Database: SALES OUTLETS
— Select. . : POPULATION > 500000 OR INCOME > 9500
— Calc A. . : SALESYTD * 1.25
— Sort per. : STATE,SALESYTD

—ST	...CITY...	POPULATI	..INCOME..	.SALESYTD.	.....Calc A
CA	Sacramento	275741	10326	28460	35575.00
	San Franci	678973	11741	72416	90520.00
	San Diego	875504	9908	120475	150593.75
	Los Angeles	7127432	11547	164537	205671.25
		8957650	43522	385888	482360.00
IL	Chicago	3005072	10455	129830	162287.50
		3005072	10455	129830	162287.50
MA	Boston	562994	9351	65823	82278.75
		562994	9351	65823	82278.75
NY	Buffalo	290183	9613	38291	47863.75
	New York	7582291	9839	192831	241038.75
		7872474	19452	231122	288902.50
TX	Austin	354496	12146	60305	75381.25
	Dallas	903078	9932	113052	141315.00
	Houston	1594086	10638	173827	217283.75
		2851660	32716	347184	433980.00
TOTALS:		23249850	115496	1159847	1449808.75

11 rcds. End of list.

TYING IT ALL TOGETHER.

Being able to visualize your information graphically is also important because it allows you to make comparisons and spot trends more easily.

UltraFile has a very easy-to-use, yet sophisticated and accurate "graphing" system. And it's fun to use, too.

Suppose you'd like to see the sales report information for cities in graphic form to make a visual comparison. Just "point" to the kind of graph you want to see, whether it's a 3-dimensional bar, circle or area graph (to name a few), and

UltraFile:™

ment tool you'll ever need.

UltraFile does the rest. It will find the information, do the calculations and create the images for you. In seconds.

But one of the best parts about UltraFile's graphing capabilities is the ability to actually do calculations right on the screen.

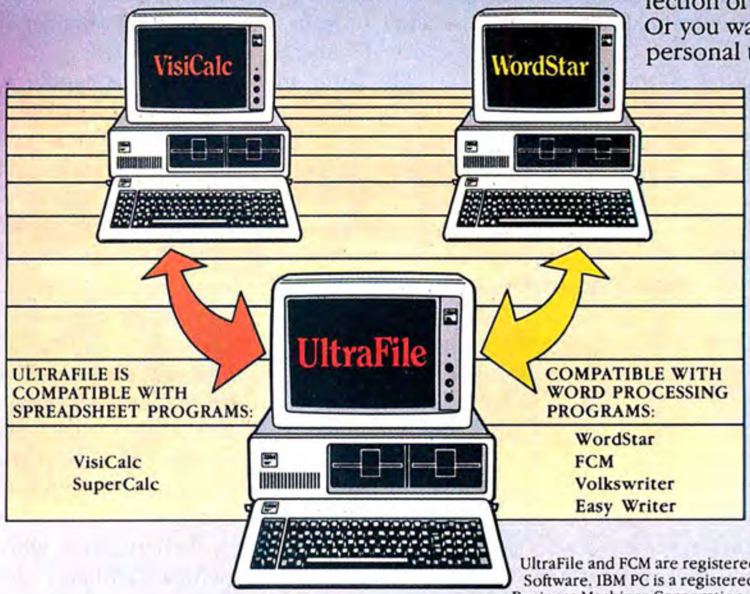
Take the average sales report information: You can pinpoint the area you're interested in and get an exact translation into numbers from where you are on the graph.

For example: If you'd like to find out what a 25% increase in sales would mean in those cities which have a population of at least 500,000 or average income of $9,500, just press a few buttons and the graph will change before your very eyes. It's that easy.

3 PROGRAMS FOR THE PRICE OF 1.

When we developed UltraFile, we put Filing, Reporting and Graphing into one easy package. So you not only get three programs, you get them for one price.

File, Report, Graph.
UltraFile: $195 complete.



ULTRAFILE ADDS NEW DIMENSIONS TO YOUR OTHER SOFTWARE PROGRAMS.

If you think UltraFile sounds terrific, you're right.

But there's more.

We taught UltraFile to talk to the most popular word processing and spreadsheet programs: VisiCalc, WordStar, SuperCalc, FCM, Volkswriter and Easy Writer. And that means you get the best of both worlds. You can pull information from your "file" and put it into a letter or use it to do more complex calculations on a speadsheet. And vice versa.

UltraFile gives your spreadsheet and word processing programs a lot more muscle.

The sales manager's example is just one way of solving one problem. UltraFile can just as easily and efficiently solve *your* particular information problems.

The point is, you create your own headings with your own information and UltraFile does the rest. Whether you want to know the total value of your inventory or which items need to be reordered. Or you want to find out what would happen to your profit if you increased the price of item "A" by 10%.

Or if your need sales figures for last quarter from all clients in the western region. Or you want to catalog your vast collection of Chinese recipes. Or you want a list of your personal tax deductions.

The list is endless. And UltraFile can do it all for you.

In fact, UltraFile gives you dozens of good reasons for owning a computer.

Stop by your Continental Software dealer. Or call us today to find out more about UltraFile and how it will help you get a grip on information.

UltraFile and FCM are registered trademarks of Continental Software. IBM PC is a registered trademark of International Business Machines Corporation. VisiCalc is a registered trademark of VisiCorp Personal Software. WordStar is a registered trademark of MicroPro International, Incorporated. SuperCalc is a registered trademark of Sorcim Corporation. Easy Writer is a registered trademark of Information Unlimited Software, Incorporated. Volkswriter is a trademark of Lifetree Software, Inc.

UltraFile is available for the IBM PC and IBM PC XT.

ULTRAFILE:
Get a grip on information.

Continental Software

A Division of Arrays, Inc.

Continental Software
11223 South Hindry Avenue
Los Angeles, California 90045
213/417-8031, 213/417-3003

COMM LINES

BY KEVIN GOLDSTEIN

Sending a disk file from one computer to another over a telephone line involves changes—lots of them.

The principal problem in sending digital data over telephone lines is that computers and telephones use incompatible types of electrical signaling schemes. Computers are digital devices, and telephones are analog. But just what does it mean when we say "analog signal," and if the analog and digital worlds are so different, how is it that we can use the telephone line—an analog transmission medium—to send digital data?

In order to answer that, let's step back for just a minute and look at a few characteristics of electricity.

To get anything done using electricity, we have to have some force capable of pushing electrons around; we get that force by harnessing a source of energy. This is a concept you're already familiar with. A car moves when the chemical energy in gasoline is released and used to force a piston down. Before it's burned, the gasoline is said to have potential energy. If we raise a bucket of water, we are increasing its potential energy; we could make use of this energy by dumping the bucket on a water wheel, thereby turning the potential energy into some useful work.

Living Better Electrically. The situation with electricity is analogous; we need some potential energy that we can harness if our electrical thingamajig is going to get anything done. Interestingly enough, just about the easiest energy source to tap for this purpose (and historically the first to be tapped) is also chemical in nature. It was discovered during the first half of the nineteenth century that certain chemical reactions could be used to drive electrons along a wire from one place to another. This discovery gave us the battery.

The energy stored in the chemical bonds of the materials used in a battery, just like the energy stored in the chemical bonds that hold together the hydrocarbons in your gas tank, is potential energy that can be made to do useful work. The chief difference between the energy stored in a battery and the energy in your gas tank is the manner in which it's liberated and harnessed.

The force that a battery can apply to pushing electrons around is a physical property intrinsic to the material in the battery; it goes by the formal name of electromotive force. You probably know it as voltage. A volt is the unit used to measure electromotive force, just as the second is the unit used to measure time.

Okay, so we've got some energy stored in a battery, and we can use that energy to force some electrons here and there, and we call that force electromotive force and measure it in volts. And what that force is trying to do is push electrons out one side of the battery and pull them in the other. Where should we send our electrons?

Since we've got a battery, and since everybody is familiar with a flashlight, why not force those electrons through a little light bulb. Simple—you know how to do this. We'll use a piece of copper wire, because one of the physical characteristics of copper is that it offers little resistance to the flow of electrons—that is, it takes very little force to push electrons through a copper wire. We can use copper to connect one end of our battery to one terminal of a light bulb, and we won't have to worry about unnecessarily wasting energy getting electrons to the bulb.

We can use another piece of copper to connect the other terminal of the bulb to the other end of the battery. And what's inside the light bulb, connecting those two terminals? Simply a piece of another kind of wire, usually tungsten. Unlike copper, tungsten is not conducive to the flow of electrons; it takes a relatively large amount of force to push electrons through this material, so when we complete our electrical path (called a circuit), letting electrons flow out one terminal of the battery, to the bulb, through the tungsten filament, and back to the other side of the battery, most of the work will be expended in pushing the electrons through the tungsten in the bulb.

When we burn gasoline, we see the results of energy being released (or used, or expended) in the form of light and heat; when we run electrons through a light bulb, we again see the results of energy being expended as light and heat—this time from the bulb.

When we connect a battery to a light bulb, we're using electromotive force to push electrons through a medium—the tungsten filament—that offers resistance to the passage of electrons. A medium that offers resistance to the flow of electrons in an electrical circuit is called, conveniently enough, a resistance. The unit by which resistance is measured is the *ohm*.

The electromotive force of our battery is measured in *volts*; if we apply a battery of higher voltage to the same resistance (the tungsten filament), we'll push more electrons through the bulb. In other words, we'll have a higher volume of current. The unit by which current is measured is the *ampere*, commonly abbreviated as amp.

Ohm's Law. By now you can probably figure out Ohm's Law, the equation that describes the relationship between voltage, current, and resistance. If you increase the electromotive force and don't change the resistance, the current increases. That means that voltage is directly proportional to current:

voltage = current × some constant or other.

It also follows that if you increase the re-

ACCOUNTING SOFTWARE

for Portable, Personal and Desktop Computers

Available for these popular 8-bit and 16-bit microcomputer formats:

All CP/M® 8" Computers (SD)
All IBM PC Compatibles
TRS-80 Model 2 (with CP/M)
TRS-80 Model 4 (with CP/M)
Apple II (with CP/M)
Apple IIe (with CP/M)
Franklin ACE (with CP/M)
Morrow Micro Decision
Northstar Advantage
Corona PC & Portable
Corona Portable
COMPAQ Portable
Hyperion Portable
Seequa Chameleon
TI Professional
Columbia MPC
Osborne Executive
Osborne I (SD)
Osborne I (DD)
DEC VT-180
CompuPro
Xerox 820
IBM PC
IBM XT
KayPro II
KayPro 10
Call about other formats

SATISFACTION GUARANTEED or Your Money Back

Complete Package of Software, Training Aids, Manuals, and Users' Newsletter

ONLY

$395
CP/M format

$495
IBM format

"...for several years the defacto standard for microcomputer software, and still a good example of some of the better thought-out software on the market." —Interface Age

Desktop Accountant™

FULL-FEATURE, FULLY-INTEGRATED ACCOUNTING SYSTEM

Everything you need to keep the books...at a price you can afford!

Your Bookkeeper, Accountant, and Banker will love you for installing this fine system!

Check these features:

This remarkably-valued Accounting System will manage your company's business records and **automate your entire bookkeeping process**—from the posting of individual transactions to producing up-to-the-minute Income Statements, Balance Sheets, and other important management reports. Desktop Accountant™ gives you a "Big Company," professional image with **printed** invoices, statements, and checks. It is **user-friendly, completely menu-driven,** offers **system-prompted data entry,** and is compatible with both floppy and hard disks. Feature-for-feature, and dollar-for-dollar it is **the best software value on the market!** But don't let the **low price** fool you. Desktop Accountant™ is **not a "cheap" accounting system, only inexpensive.** That's because original development costs were recovered years ago allowing us to sell **an excellent product** in high volume at low prices and still earn a fair profit. This **high-quality, reliable software** was designed and written in 1975 for small-to-medium sized businesses. Since then, **it has successfully proven itself in well over 10,000 companies** in all 50 states and in 13 foreign countries. It is also used and recommended by **hundreds** of bookkeepers, accountants and CPA's, including the regional offices of several famous "Big Eight" accounting firms! No other Accounting System offers you so many features, benefits, and valuable extra services. You can pay more but you can't buy more! Desktop Accountant™ is backed by our 30-day, no-risk, **money-back guarantee** of 100% customer satisfaction! Call for **free literature** or order now with complete confidence.

Read What Users Say...

"Two hours after receiving the Desktop Accountant I was up and running live data, and generated an entire month's work and financial statements the following day."
—J.C. Bartels, President
(accounting firm) Gonzales, Texas

"To get anything better we would have had to spend multiple thousands of dollars. I think it's a dynamite package for the money."
—Bob Cox, V.P. General Mgr.
(manufacturing company) San Antonio, Texas

LETTERS ON FILE

Here's what you get:

1. A professional-quality, full-feature, fully-integrated accounting system. **2.** Six-to-fourteen diskettes (depending on your system's disk storage capacity). **3.** Four easy-to-follow, self-teaching User's Manuals (over 540 pages!) that are **highly rated by satisfied users. 4.** An exclusive, new **audio cassette tape** with **step-by-step instructions** for quick installation and start-up. **5.** "Read Me First" a unique 40-page system guide and start-up aid. **6. Complete sample data and sample reports** for practice and training...a big help to new users. **7. FREE 1-year subscription to Users' Newsletter,** "Debits and Credits." It's full of tips, ideas, news, and useful suggestions from us and other users. Recent issues have run over 20 pages long! **8. FREE complete Source Code** in popular MBASIC so you can easily customize *any* program to suit your special needs. **9.** Telephone "Hot Line" service for helpful Installation support, Start-up support, Technical support, and Training support. The quality of our cheerful customer support has been acclaimed **"excellent," "first-rate,"** and **"very courteous and professional"** by our users. **10.** Access to our growing network of **Experienced User Consultants**™ in both the U.S. and Canada. These qualified, knowledgeable businesspeople themselves own this fine software and are available to assist you with almost any aspect of installation, training, operation—or even customization, if desired. You'll receive the latest **Directory** of these helpful Consultants.

Look who's using it:

Here is a sample (taken from our customer files) of the wide variety of businesses successfully using this time-tested, solidly-established, user-proven software:

• Bookeeping services • CPA/Accounting firms • Construction companies • Wholesale Florists • Retail Florists • Printers • Insurance Agencies • Shopping Centers • Machine Shops • Engineering firms • Software Companies • Computer retailers • Publishers • Financial services companies • Investors • Appliance service firms • Dairies • Boilerworks • University Business School • Catholic Diocese • Baptist Ministeries • Sail Maker • Convalescent Hospital • Car Wash • Public Library • Cement Plant • Industrial products manufacturing companies • Mortuary • Coal Company • TV/Film producers and distributors • General Store (in Alaska!) • Health services organization • Government Agencies • Record and Tape stores • Trade Association • Graphics firms • Office products companies • Real Estate firms • Oil Products distributor • Instrument Company • Hydraulic Products company • Outdoor Expeditions company • Many types of schools • Chiropractors • Physicians • Landscapers • Law firms • Restaurants • Plumbing Companies • Railroad • Farmers • Banks • Consultants • and many, many, more!!

*Call for **FREE LITERATURE,** including sample reports, comparison chart, user letters, etc.*
Not sold in stores • Order today by Mail or Phone • Phone Orders Filled Promptly

Orders, Questions, and Literature Requests CALL NOW, TOLL-FREE

MasterCard VISA **1-800-832-2244**
In California call 1-800-732-2311
Technical Support Hotline (8 AM to 5 PM, Mon.–Fri.): (415) 680-8378

Please mention Ad #120-H

Send Orders To: 1280-C Newell Avenue, Suite 147-H, Walnut Creek, California 94596

• When ordering please mention or include the Ad number appearing near our telephone number • Orders shipped within 48 hours via UPS • Add $5.00 for faster 2-day Rush Air Service • California residents add 6½% Sales Tax • Payment by VISA/MasterCard/COD/MO/Cashier's Check • All Brand Names are manufacturers' registered Trade Marks • No sales to Dealers • Foreign orders please call or write before ordering •

A Complete System!

includes all four standard general accounting programs:

GENERAL LEDGER is completely integrated to automatically post end-of-month transactions from A/R, A/P and Payroll. It **prints 13 detailed reports** for the entire company, and optional **Income Statements for up to 99 departments.** It produces up-to-the-minute **comparative financial statements** with current, year-to-date, budget, and last year (month and YTD), showing **both dollar amounts and percentages,** and presents everything you, your bookkeeper, and your accountant need to know about the company ● G/L **automatically reconciles all accounts,** and maintains extensive, **detailed audit trails** showing the source of each G/L entry for quick, easy, tracing ● Trial Balance includes all transactions ● **Flexible Chart of Accounts** ● True double-entry bookkeeping ● Master File capacity: 400 Accounts ● Monthly Transactions capacity: 1,000 with 200K diskette; 3,500 with 500K diskette; 7,000 per Megabyte with a Hard Disk.

ACCOUNTS RECEIVABLE provides **instant, on-line customer account information (both current and aged),** with complete, timely invoicing, including open-item (or balance forward) and statement capabilities. It makes the entire billing process fast, easy, and efficient. It quickly identifies your overdue accounts, **helps speed collections,** helps control your cash flow, and assists you in making better financial decisions. Both detailed and summary **customer activity and aging reports** are instantly available ● Preprinted forms available for **invoices and statements** ● **Detailed audit trail** ● Maintains comprehensive customer data files ● **Produces 8 reports and documents** ● Includes automatic periodic customer/client billing option which is ideal for service businesses ● **Itemized monthly transactions** ● Master File capacity: 400 Customers ● Monthly Transactions capacity: 800 with 200K diskette; 3,500 with 500K diskette; 7,000 per Megabyte with a Hard Disk.

ACCOUNTS PAYABLE maintains complete vendor/voucher history and includes **check-writing** capabilities. **Current and aged payable reports** are available upon command. It prepares an extremely useful **cash flow/cash requirements report** that greatly improves management control of your most valuable resource—cash! It prints checks (on commercially available forms on which **your company name, address and logo can be imprinted)** with comprehensive check stubs that your vendors will appreciate ● **Produces 11 reports and documents** ● Automatic pay selection program allows payment by due date or by discount date ● **Manual and automatic checkwriting** ● Check register ● **Detailed audit trail** ● Itemized monthly transactions ● Master File capacity: 400 Vendors ● Monthly Transactions capacity: 800 with 200K diskette; 3,500 with 500K diskette; 7,000 per Megabyte with a Hard Disk.

PAYROLL is a complete, **easy-to-use** professional-quality payroll system. Be the office hero each week when the checks come out on time! This program **calculates payroll for every type of employee** (hourly, salaried, and commissioned) and **prints payroll checks** (with popular, comprehensive check stubs) with an absolute **minimum** of input. **Your company name and logo** can appear on these, too. ● Stores and reports comprehensive employee and payroll information. ● Maintains monthly, quarterly, and yearly totals for reporting purposes in **multiple states**(!) ● Offers **user-maintainable Federal, State, and local tax tables**(!). ● W-2 printing ● 941 Reporting ● **Produces 10 reports and documents** ● Master File capacity: 400 employees. An **outstanding** Payroll package!

Produces 42 Reports & Documents:

General Ledger 1. Chart of Accounts **2.** Chart of Accounts with summary dollar amounts **3.** Chart of Accounts with prior year comparisons **4.** Daily Transactions Report **5.** Itemized Monthly Transactions **6.** Balance Sheet **7.** Balance Sheet with prior year comparison **8.** Income Statement **9.** Income Statement with prior year comparison **10.** Departmental Income Statement(s) **11.** Departmental Income Statement(s) with prior year comparison **12.** Detail report for individual accounts **13.** Trial Balance Statement

Accounts Receivable 1. Daily Transactions Report **2.** Invoices (with or without preprinted forms) **3.** Statements (with or without pre-printed forms) **4.** Summary Aging Report **5.** Detailed Aging Report **6.** Itemized Monthly Transactions **7.** Detailed Customer Activity Report **8.** Summary Customer Account Report

Accounts Payable 1. Daily Voucher Report **2.** Daily Credit Report **3.** Checks with Detailed stubs **4.** Check Register **5.** General Ledger Transfer Report **6.** Cash Requirements Report **7.** Transaction Register **8.** Open Voucher Report **9.** Aged Payables Report **10.** Detailed Vendor Activity Report **11.** Summary Vendor Account Report

Payroll 1. Federal Tax Tables **2.** State Tax Tables **3.** Payroll checks with stubs **4.** Payroll Check Register **5.** Monthly Payroll Summary **6.** Quarterly Payroll Summary **7.** General Ledger Transfer Report **8.** Detailed Employee File Listing **9.** Produces 941 Worksheet **10.** Prints annual W-2 Forms

System Requirements: Either CP/M® or MS-DOS (PC-DOS) ● Microsoft BASIC ● 64K RAM ● Two disk drives or hard disk ● 132 column wide carriage printer, *or* an 8½"x11" printer with compressed print mode (an Epson MX-80 or similar printer).

100% SATISFACTION GUARANTEE

No-Risk, 30-day Money-Back Guarantee

Order Desktop Accountant and evaluate it yourself for 30 days. Test the performance of each program using our comprehensive *Sample Data Files* to enter and process data, query accounts, print checks and reports, and more. Review the five User's Manuals, hear the Training Tape, read the users' Newsletter, even call our Technical Support Group with your questions. ***You must be completely satisfied*** with Desktop Accountant or you may return it in good condition *with the Sealed Master Disk UNOPENED* (it installs *your* live data) for a prompt, courteous, complete refund!

ROCKY MOUNTAIN SOFTWARE SYSTEMS
Specialists in Accounting Software

sistance but keep voltage the same, the current will drop. So current is inversely proportional to resistance:

current = 1/resistance × another constant.

Finally, if you increase the resistance but want to keep the same current, you must increase the voltage; therefore voltage is directly proportional to resistance:

voltage = resistance × yet another constant.

Now, there is one equation that is consistent with those three equations:

current = voltage/resistance × some fourth constant.

If we define the units of current, voltage, and resistance in such a way that the fourth constant is equal to 1, then our equation becomes:

current = voltage/resistance.

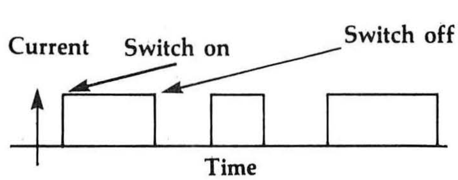
which is indeed Ohm's Law. Just for the record, current is frequently represented by the letter I, voltage by the letter E, and, for some reason, resistance by the letter R; so you'll often see Ohm's Law stated as

$$I=E/R.$$

Don't worry about it; it says the same thing as the wordy formula above (the scientific elite is just protecting its turf).

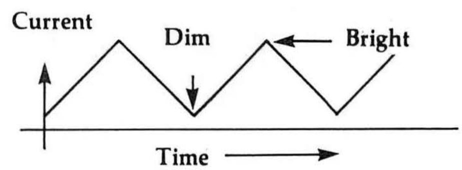
Now that we understand the three basic properties of an electrical circuit, it's a fairly easy matter to define the difference between analog and digital signals. Here's a quick definition, and then we'll look at some examples to see just what our definition means in real life: A digital circuit is one in which current changes rapidly between two distinct values; the magnitude of the current will always be at one of those two values and never in the middle (with the exception of the very brief time period during which the signal is switching from one value to the other). An analog circuit is one in which the current may take on all values in a smooth and continuous manner.

So how do we generate a digital signal? Well, if we put a switch in our light-bulb—battery circuit and flip it on and off, we'll have a digital circuit; the current through the bulb will always be either zero (switch open, light off) or some value defined by Ohm's Law (E/R, light on). A picture of the current through the light bulb at different times might look like this:

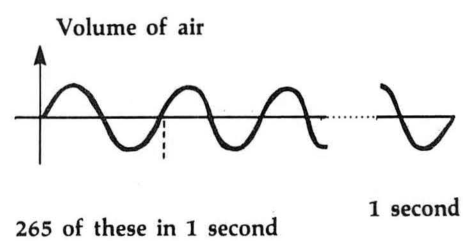


Since the diagram looks like a wave that's been squared off, it's called a square wave. And since the drawing shows the relationship of an electrical property over time, it's called a timing diagram.

Analog signals are much more prevalent in the real world than digital signals, although it's a little harder to come up with a simple example. If we could magically vary the length of the filament in the light bulb (and thus its resistance) smoothly and continuously between its two endpoints, we would have an analog signal; the light from the bulb would vary smoothly between very dim and very bright. If we changed the length of the filament regularly over time from its maximum to its minimum and back to its maximum, the current passing through the bulb would look like this:



Analog in Middle C. Your voice is an analog signal, although not an electrical one: You can vary both its loudness and frequency over a wide continuum. Now, let's say one of you is blessed with perfect pitch and can hum middle C for us. A sound wave is just air being pushed back and forth; if we could measure the amount of air being pushed by your humming, we would get a drawing that would look something like this:

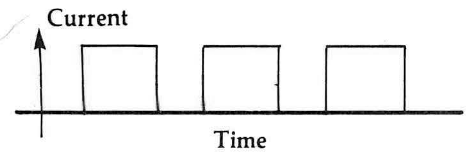


Before you start humming, no air is being moved; as you start, you begin moving air in one direction up to a maximum; then the amount of air being forced in that direction slowly falls, until you are pushing no air at all. Then your larynx starts moving air in the other direction, again working smoothly up to a maximum in that direction, then falling back toward zero. You may recognize the previous diagram as a sine wave.

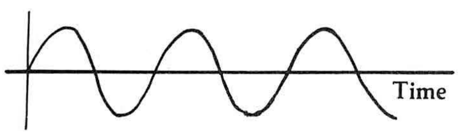
Now voice is what Bell intended to send across phone lines. It's pretty simple, using a microphone, to turn that air wave into an electrical signal that varies at the same rate, and that analog signal can be put through a phone line.

But here's something interesting. If we start flipping the switch like mad on our

light bulb—battery circuit at very regular intervals, the digital square wave will look like this:



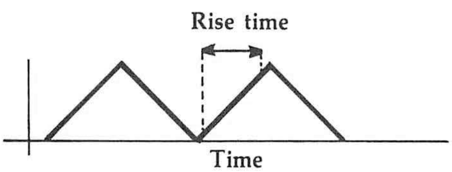
If we compare that to our analog signal:



we'll see the two look somewhat similar. Indeed, if we could square up the sine wave and move the whole thing up, or chop off the corners of the square wave and move it down so it centers about the x-axis as the sine wave does, the two waves would be very similar. Since by now you've probably realized that the digital light-bulb signal is very similar to what computers put out, you may be asking why we can't just connect that square wave directly to the phone line, since it is fairly similar to the sine wave that we know is okay on a phone line.

Because all hell would break loose, that's why. At the very least, your neighbors would never talk to you again. You see, the faster a signal changes, the more likely it is to generate noise in nearby (and not so nearby) conductors. (Such as the telephone wiring in your neighbor's house, or worse, the antenna for your TV.) Analog signals change relatively slowly and therefore generate relatively little interference (noise, crosstalk, or static). The *rise time* of a digital signal—that's the length of time it takes for the signal to change from (close to) its minimum to (90 percent of) its maximum—is, on the other hand, very small, which means that digital signals are quite likely to generate garbage in other devices.

Here's a graphic explanation of rise time:

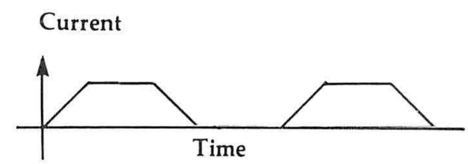


The FCC and Interference. The fast rise times characteristic of computer circuits are the reason the FCC has gone out of its way to make sure that home computers comply with interference regulations; it's also probably why your IBM pc has a metal rather than a plastic case.

Some time ago, the Electronics Industry Association, a standards-making body of the electronics industry, published what is known as the RS-232-C interface standard. This standard defines, among other things, the range of voltage levels and impedances (a fancy word for a total resistance consisting of many components) within which a circuit must stay to be compatible with other RS-232 interface circuits. What the RS-232 standard does not govern, however, is signal rise time.

The military has a standard almost identical to RS-232, with one major difference: The military version specifies a maximum rise time. That's not only because the military wants to avoid generating interference in other critical circuits, but also because it's difficult to prevent "leakage" of digital signals with fast rise times; in other words, it's pretty easy for someone to read a digital signal with an electronic snooper, even from some distance.

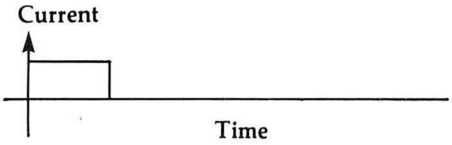
The mil-spec version of the digital signals above might look like this:



Apart from problems relating to interference, fast rise times are in general a good thing; signals with slow rise times can get sloppy and hard to decipher, limiting the maximum rate at which data can be transmitted.

Back to putting our digital signal directly on the phone line. Obviously, we have to round out those corners first; if we did that, could we connect more or less directly to the phone line?

Well, yes. And no. Yes, if all we ever wanted to send were characters similar to the character U, which alternates between bits that are on and off and thus looks just like that square wave above. (Those of you saying, "Ah, that would be true, except for the start and stop bits," are ahead of the story. The rest of this discussion will conveniently ignore framing bits; they'll be explained later.) But suppose we wanted to send an at-sign (@), which has its first bit on and the remaining six bits off. If we sent those bits in order, one after the other, the timing diagram would look like this:



Or suppose we wanted to send a bunch of smily faces which consist of six bits off followed by one bit on.

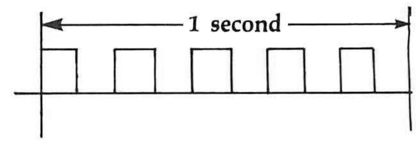
Could we send those?

To answer those questions, the first thing we need to do is look closely at some of the limitations of the telephone transmission line. How big is the transmission pipe? How much data can we stuff through it, and how fast? What other limitations does a phone line have that we should be aware of?

A Hertz Is Born. Look for a second at the diagram of air volume versus time for our hypothetical middle C hummer. The reason middle C sounds like middle C is because the air flow goes through a complete cycle of zero-to-maximum-to-zero-to-minimum-and back-to-zero 265 times per second; 265 is said to be the frequency of the note called middle C. If our hummer hummed into a microphone and generated a similar-looking electrical signal, a current with a frequency of 265 cycles per second would be generated, and until not very long ago that's exactly how its frequency would have been labeled: f=265 cycles per second. Since that was too clear, the powers that be decided that the new name for cycles per second should be Hertz; one Hertz (1 Hz) is one cycle per second. (In all fairness, the new nomenclature is more succinct.) Thus the signal generated by middle C is 265 Hz.

The range of the typical human voice is about 300 to 3,000 Hz. Since the phone was made to pass human voices, the range of frequencies it can pass is also limited to approximately 300 to 3,000 Hz. The range of frequencies that a transmission channel can pass without distortion (meaning without altering the signal beyond some predefined limit) is called its *bandwidth*.

Bandwidth is a measure of the information-carrying capability of a transmission channel. Let's assume we're trying to pass a digital signal that consists of a string of alternating 1 and 0 bits. And let's assume our machine is very slow—that it puts out only ten bits per second. Let's draw it.



As you can see from the diagram, even though our machine is cranking out ten bits per second, there are only five complete cycles of 0-to-1-to-0; thus our signal effectively has a bandwidth of only 5 Hz. Could it be put across the phone line? Nope. The phone line has a minimum frequency of 300 Hz.

But how about if our computer cranked out bits at two thousand bits per second? That should result in a bandwidth of only 1,000 Hz., nowhere near the cutoff of 3,000 Hz. Could we put that across the phone line?

Theoretically, yes—if we could round off those sharp corners. But if we went back to sending at-signs, we'd see that one complete 0-to-1-and-back cycle would occur only one-seventh as often; when sending a series of at-signs, the bandwidth drops to 2000/7, or 285 Hz., and that's below the minimum frequency of the phone line. So we've obviously got a problem: It appears that if digital data is connected directly to a phone line, the ability of the line to transmit the data varies with the data being sent. Not a good situation.

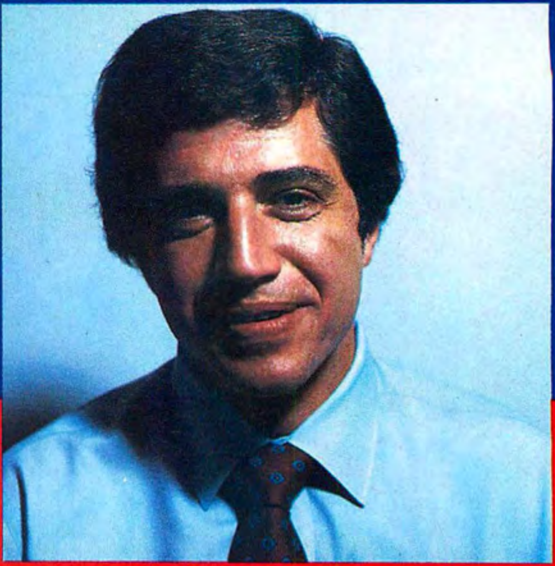
As if those problems weren't enough, we'd run into a few others if we tried to connect two computers directly to the phone line. First off, data bits alone don't mean much; they need to be grouped into characters to make sense. If we just send a string of bits one after the other, how does the receiving computer determine where one character stops and the next one begins—in other words, how does it locate the character boundaries? And suppose while receiving a file we realize we're about to run out of disk space. How do we signal the sending computer to suspend transmission without our signal corrupting the data we're receiving?

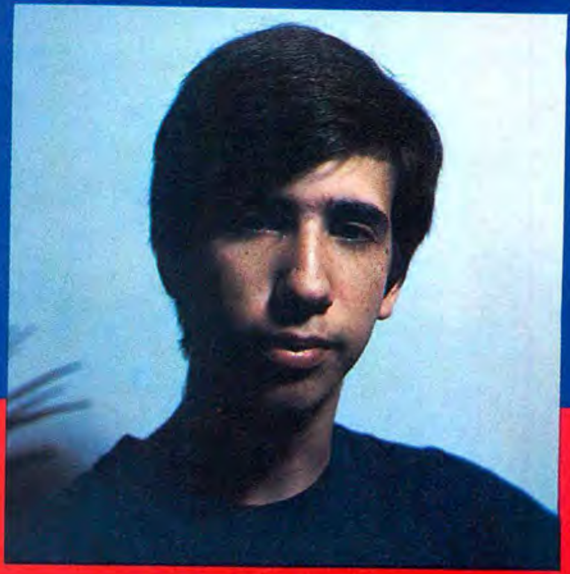
By now you have a good idea of the kinds of problems that need to be handled by telecommunications gear. Some of those problems are tackled by the modem; others are handled within the computer by hardware on the asynchronous board, or by software running on the CPU itself. We'll look at those solutions in future installments. ▲

Exec Continental:

A SOFTWARE SURVIVOR

by David Hunter





Tall, physically fit, inwardly composed, James D. Sadlier is one of those smart businessmen who quietly go about their business and make small fortunes. He's a careful observer of the business community and he's patient. As president of Continental Software and the Book Company, he is capable of making decisions fast and acting upon them with the same speed.

Initially drawn from the East Coast to California by things other than computers, Sadlier is one of the lucky ones who struck it rich in the silicon Promised Land. He would probably have succeeded wherever he went.

Stock and Trade. Prior to entering the microcomputer field, Sadlier worked as a floor trader in the securities industry. He is a native of Brooklyn and grew up in New York City. A graduate of Brooklyn College with a degree in economics, Sadlier spent several years on Wall Street before moving to the West Coast in September 1974.

Sadlier's strength comes from a fundamental grounding in business and the experience he gained while holding a seat on the American Stock Exchange and as a partner in a small brokerage house. In the mid-seventies, as a registered representative for financial institutions, Sadlier discovered microcomputers.

While working in Merrill Lynch's institutional sales department, Sadlier researched the investment potential of Commodore and its Pet line of products. In early 1977, he traveled to the company to meet with Commodore founders Chuck Peddle and Jack Trameil. He came to a quick decision that personal computers might be the next hot industry.

"I got so hyped by the potential of the industry," he says, "that I started looking for a way to enter it."

But how does a Wall Street businessperson with no technical knowledge get into the computer industry? Simple, ally yourself with someone who has the expertise and then dazzle them with your business sense.

Sadlier saw a magazine ad for a company that was offering a complete turnkey operation for opening a computer store. That company was ComputerLand. After meeting with the firm's cofounder and president, Ed Faber, Sadlier decided that the Sunnyvale, California-based group had the best program he'd heard of for opening a retail store.

The Right Spot. When Sadlier's ComputerLand of South Bay, in Lawndale, California, opened in early 1978, it was the fourth such franchise store in southern California. It was a risky operation; computer stores started up and failed by the hundreds in the late

Opposite page (top to bottom): Jim Sadlier, president of Continental Software and the Book Company; Hank Scheinberg, Continental's executive vice president and director of marketing; Kathy Farmer, vice president of administration. This page (clockwise from upper left): Mary Watt, vice president of operations; James Densmore, special assignments; George Rodgers, vice president of purchasing and production; Rich West, customer service manager.

Softball player, retailer, software publisher Jim Sadlier.

seventies and early eighties. But Sadlier chose his location wisely—a strategic spot south of Los Angeles, near affluent Palos Verdes, industrial parks, and high technology firms such as TRW and Northrop.

To compensate for his lack of a technical background, Sadlier relied on ComputerLand and hired knowledgeable people to work in the store. From the start, ComputerLand of South Bay was a successful venture.

When Sadlier first opened the store, it was practically a microcomputer supermarket—carrying Imsai, NorthStar, Cromenco, and Apple products. As more low-priced systems like the Apple II came out, the technical hobbyist community was drawn to microcomputers. Later the market would shift to small business and would fast approach the mass consumer stage.

The shifting market trends of the microcomputer industry, reflected in the day-to-day business transacted at his store, were something that Sadlier was trained to watch. With his foot in the door of the industry, Sadlier began peering inside and gathering information.

Above all else, he listened to the customers at his store. What did they like and dislike? What did they need and want? What did they not understand? After a couple of years of observing, Sadlier believed he knew the answers to some of those questions.

Sadlier identified two areas that caused customers the most problems—a lack of good information about software, and the fact that there was little good software to begin with. In 1980, he began thinking of ways to address these needs of the market.

A Local Meeting Place. ComputerLand of South Bay was much more than just another outlet for microcomputer products. It was a meeting place, a temple for the lost and weary in the early days of personal computing. Sadlier became a wise leader, and some of his customers formed a skilled and loyal following.

When he began developing his first product, the 1981 *Book of Apple Software,* Sadlier naturally turned to his more knowledgeable customers for contributions to the collection of software reviews. One of these was Jeffrey Stanton, a programmer and general computer whiz. Stanton wrote reviews for and edited the first *Book of Apple Software.*

Two early customers, Kathy and Mike Farmer, became loyal disciples of Sadlier. One day they came into the store and listened to Sadlier complain at length about what a hassle all the store's paperwork had become. "I told him I *love* to do paperwork," says Kathy Farmer. "He said, 'You're hired.' "

Sadlier, Farmer, Stanton, and a high school student, Pam Nowatka (who worked part-time in the store), shipped the first copies of *The Book of Apple Software* out of a storefront located next to the South Bay ComputerLand. It was a modest beginning. Then, Sadlier turned his attention to software.

Bob Schoenberg and Steve Pollack were two would-be programmers when they first met Sadlier through the ComputerLand store. Sadlier had noticed that Apple's *Checkbook* program seemed to sell well. He could never stock enough copies of it in the store. He also remembers listening to the comments of users of that program.

"They said, 'I wish it could do this and I wish it could do that,' " recalls the nontechnical Sadlier. Well "this and that" is what Sadlier, Schoenberg, and Pollack agreed upon as the basis for the *Home Money Minder*—the earliest precursor of *Home Accountant*—which sold for $34.95 in late 1980.

Twins. Sadlier formed the Book Company and Continental Software about the same time. Both were risky propositions, perhaps more risky than opening a computer store. Even more challenging was trying to make the transition from seller to producer—a transition few have managed successfully.

Sadlier decided to give it a go and experiment. He reasoned that listening to customers and following his business sense and the advice of individuals like Softsel's founder Bob Leff (Sadlier first met

Leff back when the now-giant distribution company was being run out of Leff's apartment) ought to ensure success—it has, and Sadlier is not surprised.

"Could I imagine the success Continental would have? Yes. I've looked at the historical perspective, the life cycles of many different industries. I know how fast they can explode and grow. Computers are just like space technology. I've seen it a lot of times."

That is not to say that it's been easy to start up the two companies or that Sadlier didn't do a lot of experimenting in the beginning. His first five software products were a curious mix of the serious and the frivolous—*L.A. Land Monopoly, The Mailroom, Hyperspace Wars,* the *Home Money Minder,* and *General Ledger.*

In those days, the game market was not dominated by a half-dozen or so companies the way it is now. Sadlier, like a hundred other software publishers, tried his hand with limited success. *Monopoly, Hyperspace Wars,* and later games like *3-D Skiing, Guardian,* and *Cross Country Rallye* (all for the Apple) were grand experiments, but Sadlier knew when he was outclassed. Since the summer of 1982, Continental has pretty much left games to the games companies.

Sadlier is first and foremost a businessman, so it's not surprising that Continental has concentrated almost entirely on financial and business software for the home and the small business. What's surprising is how well the company has succeeded with only a limited number of products. Credit goes to Sadlier and his business philosophy shared by those who work closely with him.

Sadlier cultivated the art of listening to the buyers of *Home Money Minder* and *The Mailroom,* just as he had listened to customers in the retail store. He believes the relationship with the customer is crucial, for this is the best way to find out what is lacking in a product. And improving a product is the best way to keep making money.

Home Accountants. In late 1981, Schoenberg and Pollack teamed up with Larry Grodin to work on an updated *Home Money Minder,* changing the product's name to *Home Accountant. The Mailroom* became *1st Class Mail* in late 1981, and then metamorphosed again early this year into *FCM* with added features, such as the ability to merge with word processing programs.

When IBM announced its Personal Computer in late 1981, Sadlier had some initial doubts about the machine but went ahead with plans to convert his two biggest programs. Kathy Farmer's husband Mike, a programmer by trade, came in to do the conversion of *Home Accountant* to the IBM Personal Computer.

Few can argue with the success the program has enjoyed in the pc market. It has been on every Top Thirty Bestseller poll, usually in the top ten, since *Softalk for the IBM Personal Computer* started keeping score fourteen issues ago.

Perceiving a need for a good tax program, Sadlier brought out *Tax Advantage* late last year. Although it didn't outsell more established products, *Tax Advantage* made a decent showing in both the Apple and IBM markets. You can bet that Sadlier and company are listening closely to the program's owners, planning for next year's update.

The IBM Personal Computer has been good for Sadlier's ComputerLand store and for his software publishing company. In retrospect it was only what he expected.

"The only thing that surprised me," says Sadlier, "was how fast IBM reacted and brought out the machine. It was a delightful surprise. Working in a time frame, they addressed the issue. They cut through red tape, committed the necessary resources, and did it, pulled it off.

"They violated traditions, but they did their homework. I think the Personal Computer is one of the best success stories of a major corporation in recent years."

Dynamic Duo. Two years ago, the Book Company and Continental Software were still fairly small-time operations. Once *Home*

Accountant and *1st Class Mail* started to sell in large numbers, the two companies began to burst the confines of the storefront next to the ComputerLand.

Sadlier moved Continental and the Book Company into a momentarily spacious office suite near the Los Angeles International Airport. The two companies, seemingly overrun with talented and qualified personnel, are now bursting the walls of that location and are actively seeking more space.

One reason the company needs more space is because of the efforts of Sadlier's partner Hank Scheinberg, executive vice president and director of marketing for both Continental and the Book Company. Scheinberg and Sadlier have known each other for close to ten years. They have common roots—Wall Street—and the same quick-as-a-Minuteman business instincts.

A graduate of Trinity College in Connecticut, Scheinberg was raised in a small farm town in rural eastern Long Island. He spent sixteen years on Wall Street in institutional sales and research. Sadlier says that Scheinberg is good at spotting trends, at "knowing what's going to be hot."

Scheinberg moved to southern California in the early seventies and got into a hot industry—real estate. He continued investing and eventually became an executive of one of the largest real estate firms in the state. Scheinberg met Sadlier on a tennis court in Marina del Rey back in 1974 and the two have been friends ever since.

Once again, Scheinberg is certain that he's gotten into a hot industry. Though he is another nontechie, Scheinberg knows plenty about marketing. He has done a first-rate job, quadrupling Continental's sales while competing with the best of them—Apple Computer, Broderbund, and VisiCorp.

Scheinberg feels that computers are going to be a long trend (he was into real estate for seven years). He quotes John Naisbitt, author of *Megatrends*: "It isn't the Hula-Hoop."

According to the Book. The managerial role of minding the Book Company has been taken on by Dr. Robert Wells, who has been with the company for about a year. Wells is responsible for overseeing the Book Company's growing line of publications catering to several different personal computers.

One of the biggest tasks facing him and his staff is wading through the large amounts of information for the annual updates of *The Book of Apple Software* and its sister publication *The Book of Atari Software*. Wells and his staff are currently busy compiling information for the first *Book of IBM Software*, due out later this year.

The Continental Software crew is a healthy mix of the young and the old, the technically proficient and the professionally competent.

Jim Densmore, or J.D. as he's called by many, is a sharp individual in charge of a very important facet of Continental's operations. His title is special assignment; what he does is a little of everything.

Of primary importance is Densmore's overseeing of the alpha and beta testing phases before a product is released. He is one of the first to be fully aware of the quality of a product and interfaces frequently with the customer support and marketing departments. Densmore also works with authors, developing programs and enhancing existing products. He sees himself in an integral position, responsible for ensuring the quality of a program before it hits the market.

Just this past May, Gerald Lewis was appointed Continental's director of software development. He screens and evaluates new programs being considered for publication. Prior to joining Continental, Lewis was an independent systems analyst and software documentation consultant for such companies as Standard Logic.

Lewis, Densmore, and marketing manager Denny Mosier are the three one-person departments at Continental. According to the official press release, Mosier is responsible for research, advertising, public relations, planning future software programs, and supporting existing products.

Master Marketer. Mosier joined Continental early this spring; in the past, he's been an account executive with Simon/Public Relations, a high-tech public relations firm, and an L.A.-based correspondent for *Electronic News*, a trade publication.

Mary Watt is Continental's vice president of operations. Watt left a career in banking for an entry-level job in the computer industry. She found it at Continental in June of 1981. In those early days, Watt, Kathy Farmer, and Pam Nowatka for the most part ran the whole office—the business and administrative side of Continental and the Book Company.

Currently, Watt is responsible for overseeing customer service and the all-important day-to-day order entry operations. Nowatka works in accounts receivable and has the title of assistant to the vice president of administration. She can still recall that first shipment of *The Book of Apple Software* in 1980; she remembers when Continental Software was "a gleam in Jim and Kathy's eyes."

Like Sadlier and Farmer, Nowatka seems to have adjusted well to Continental's continued growth. The company now has close to sixty full-time, in-house employees, as well as a fluctuating number of temporary helpers. "We add and delete people as projects come and go," says Nowatka.

Continental's vice president of purchasing and production, George Rodgers, has been with the company since February 1982. His department of seventeen is the largest in the company, occupying a large shipping area in the back of the building. Last Christmas, Rodgers and his department went through their toughest trials. *Home Accountant*, *FCM*, and particularly *Tax Advantage* had high end-of-the-year sales, and Rodgers says the workload kept everybody jumping.

Where To Go for a Tune-up. Continental's customer service department employs seven people full-time. Customer service man-

ager Rich West says the department gets between fifty and sixty phone calls a day. On the average, one out of every twenty inquiries requires expert, technical advice. Most of the others concern simple questions that are usually answered in the manuals.

Peter Castillo runs what is called "the information desk" at Continental. A new position, the information desk is designed to relieve the customer service and sales departments. Castillo takes care of order and order-entry questions before the product is in the user's hands.

Last but not least, Continental has a crack sales staff. With over twenty-two hundred dealers nationwide, regional sales managers Tere White, Barbara Ring, and Stephanie Loysen work hard to keep in constant contact with the marketplace. They've been joined recently by sales managers Penny Olender, Jennifer Bartel, Sally Hammer, and Mike Hilton.

VisiData and Wordbase. Presently, Continental should be starting to ship its promising new database management program for the IBM Personal Computer and XT. Authored by Charles Dworkis, the product is called *UltraFile* and has some interesting features, answering what Sadlier sees as the demands of the market.

UltraFile appears easy to use, combining filing, graphics, and reports features. Given the proposed price (under two hundred dollars), *UltraFile's* unique features make it a smart product.

A strong selling point is *UltraFile's* compatibility with *VisiCalc* and *Wordstar*. The ability to merge financial and textual material with an existing database program addresses a large portion of the pc market. Sadlier's patient, but, when he makes a move, he doesn't fool around.

Sadlier is a good leader. He's serious, but approachable.

"It's always tough to find really good people," he says. "Achieving the right chemistry, a successful company personality, is a team effort."

So far, Continental and the Book Company are winning the game. A fast walk through the offices of the two companies reveals a hardworking, happy bunch of people.

If Sadlier feels much pressure being the president of two companies in the still highly volatile and highly unpredictable microcomputer industry, he doesn't show it. He's confident, for a number of reasons, that Continental and the Book Company are here to stay.

Market Watch. First and foremost is his faith in the practice of studying the marketplace and coming up with new products that meet a pressing demand. Second is his philosophy of taking existing products and improving them based on feedback from customers and dealers. Third is his belief in bringing out only those products that jive with his existing product line. It's not likely you'll see too many graphics utility or word processing programs coming out of Continental in the foreseeable future.

Sadlier is also a firm believer in being able to react fast to a changing market. It's an advantage he claims over larger companies who make "decisions by committee."

In the future, Continental and the Book Company will build on their present product lines, creating in some cases "families of products," says Sadlier. There will be many more books, software with an accounting flavor, and possibly educational software.

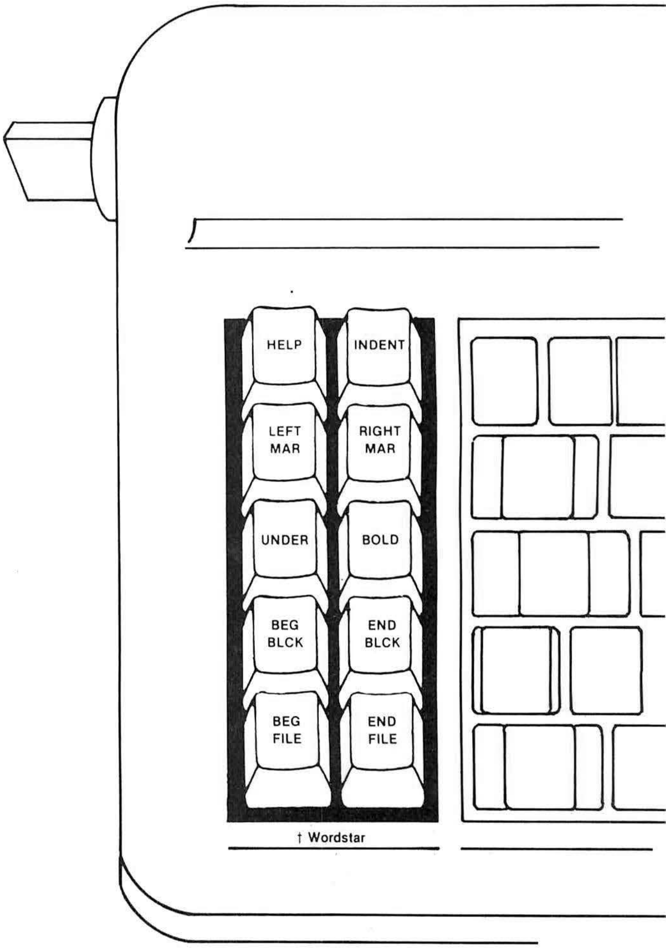
"Around the turn of the century, there were three or four hundred companies in the automobile industry," says Sadlier. "Cars became a big part of our lifestyle, and the same thing is happening with computers today.

"But that doesn't necessarily mean only a half dozen of the hundreds of computer companies around today will survive." The industry will continue to evolve; there will be a shifting of ranks from time to time and not everyone will make it. Sadlier is confident his companies can keep pace.

"We will be a survivor." ▲

KEYTOP-PC™

Transform Your Function Keys—
Bridge the Gap Between Your Software and Your Keyboard



- KEYTOPS™ made of durable and highly flexible polyurethane.
- Fits snugly on top of the 10 function keys of your IBM-PC* keyboard yet easily removable.
- Labels display function of each key depending on your application program.
- Facilitates training of personnel, avoids unnecessary delays/costly mistakes.
- Each KEYTOP™ supplied with different sets of labels that are compatible with most commonly used programs, or create your own labels.
- Packet of 5 KEYTOPS™ and sets of labels for only $19.95.
- Software manufacturers: we will customize KEYTOPS™ to your special applications.

COMPU-MED
92 Deerfield Lane North
Pleasantville, NY 10570
(914) 769-8382

* IBM-PC is a registered trademark of IBM.

† Wordstar is a registered trademark of Micropro International Corporation.

Part Three of a Series

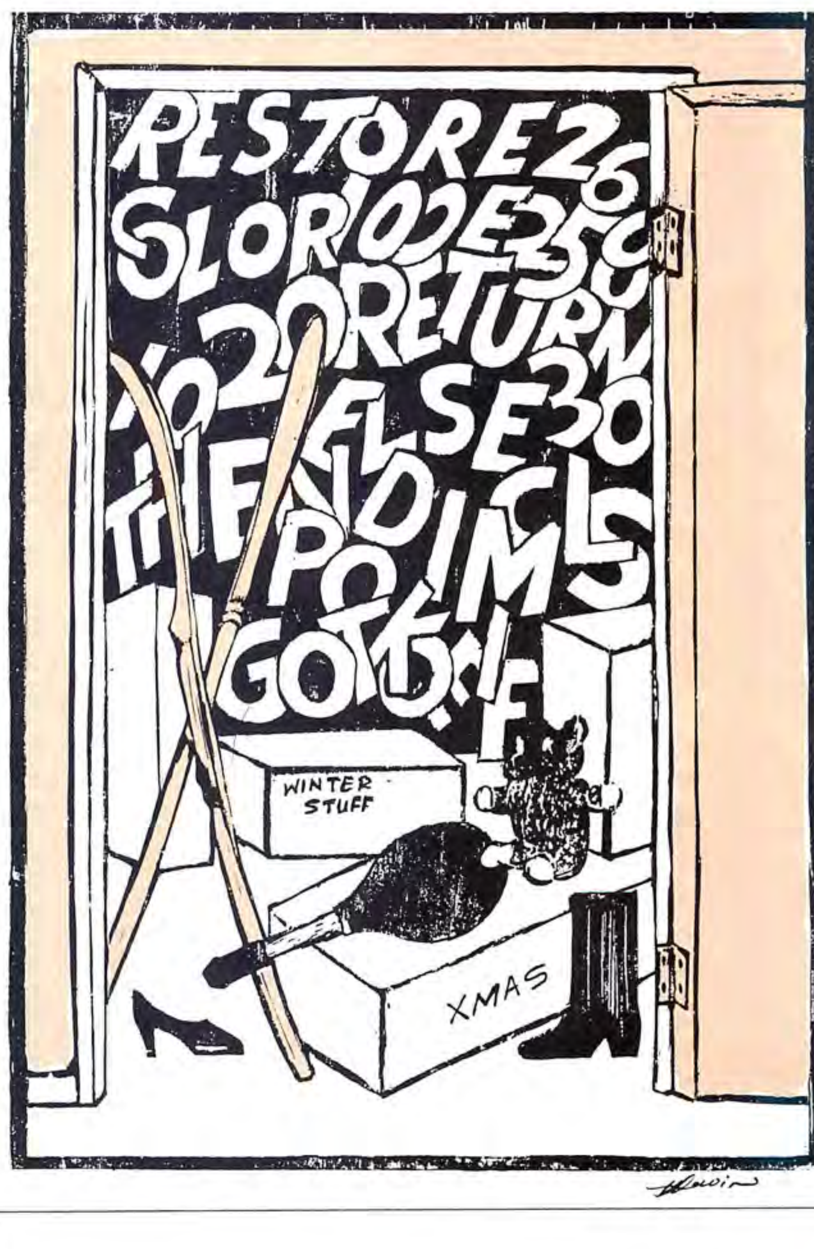
How Basic Stores Programs

by Gary Little

A pc Basic program is merely the data for a complex program written in 8088 assembly language; this complex program is commonly called the Basic interpreter. The interpreter resides partially in ROM and partially in RAM. The part that constitutes Cassette Basic is ROM-resident; the extensions added by disk Basic or Advanced Basic are stored in RAM.

The Basic interpreter does just what the name implies: It interprets the "data" that you put into memory with a load or run command (that is, your Basic program); the interpreter treats your program as a specific type of data structure and executes the series of commands dictated by the data in that structure.

You might think that your program is stored in memory in exactly the same manner as it was originally typed in from the keyboard. That's not the case, however. In order to save valuable memory space (remember that Basic programs are restricted to a 64K work space) and to speed up program execution, each line of your Basic program is analyzed and compressed by the interpreter before it's inserted into memory.



Woodcut illustration by Weldon O. Lewin

A complete plug-in communications package for the IBM PC.

Requires an IBM PC with minimum 96K bytes of memory; IBM DOS 1.10 or 1.00; one disk drive; and 80-column display. No serial card or separate power source is needed.

Smartmodem 1200B. (Includes telephone cable.)

Smartcom II communications software.

you need to "set up" your computer to match the way the remote system transmits data. With Smartcom II, you do this only once. After that, parameters are stored in a directory on Smartcom II. Calling or answering a system listed in the directory requires just a few quick keystrokes.

You can store lengthy log-on sequences the same way. Press one key, and Smartcom II automatically connects you to a utility or information service.

And if you need it, there's always "help." One of several special functions assigned to your IBM function keys, this feature explains prompts, messages, etc. to make communicating even easier.

Smartcom II also provides a directory of the files stored on your disk. Letting you create, display, list, name, re-name or erase any file right from the screen.

Like all our products, Smartcom II and Smartmodem 1200B are backed by excellent documentation, a limited two-year warranty, and full support. From us to your dealer to you.

So, see your dealer today. Link up to the exciting world of telecomputing. With Hayes complete plug-in communications package for your IBM PC.

Hayes Microcomputer Products, Inc., 5923 Peachtree Industrial Blvd., Norcross, Georgia 30092. 404/449-8791.

NOTE: The Smartmodem 1200B may also be installed in the IBM Personal Computer XT or the Expansion Unit.

In those units, another board installed in the slot to the immediate right of the Smartmodem 1200B may not clear the modem; also, the brackets may not fit properly. If this occurs, the slot to the right of the modem should be left empty.

This compression process is called tokenization, because it involves, among other things, the substitution of one- or two-byte tokens for Basic keywords. For example, if you enter the line

```
100 PRINT
```

that line will not be stored in eleven bytes of memory, as it would be if you used Edlin to create it as part of a source file (Edlin would generate nine bytes of text plus two bytes for the carriage return–line feed sequence at the end of the line). Rather, it will be stored as six bytes, in the following manner: two for the line number, one for the one-byte token that represents the keyword *print,* and three bytes of overhead; more about the overhead in a moment.

It is this tokenized version of your program, not the original source listing, that's analyzed by the interpreter. Listing a program, incidentally, is essentially a process of detokenizing, because the *list* command reads tokens and displays full keywords.

Let's take a detailed look at what happens when you add a new line to a Basic program while in direct mode—that is, when there's no Basic program running and you enter a Basic statement without a line number.

When you press the enter key in direct mode, Basic scans your input and checks to see if it begins with a valid line number. If it does not, Basic concludes that the input is a direct command and attempts to execute it. If the input does begin with a line number, Basic interprets it as a deferred command—one that's to be executed only when the program is executed; the interpreter then tokenizes the line and stores it in the appropriate position in memory.

Memory Storage Sequence. The line is placed in memory in such a way that the ascending numeric sequence of the program is maintained. The lowest numbered line is stored lowest in memory (at the location pointed to by the beginning-of-program pointer at 30H/31H in the Basic program segment), and the higher numbered lines are stored higher in memory.

A tokenized line has the following general form:

xx yy	xx yy	xx yy zz ...	00
address of next line	**this line number**	**tokens for the contents of line**	**end-of-line marker**

The address-of-next-line and this-line-number fields are stored as two bytes each, with the less significant byte coming first. The three bytes of overhead mentioned previously are made up of the two bytes allocated for the address of the next line and the 00 byte that indicates the end of the current line.

Because each tokenized line begins with the address of the next line of the program, the tokenized Basic program is a good example of a linked-list data structure. This structure is particularly well suited for the purpose; it allows the interpreter to find lines quickly and it facilitates the addition and deletion of program lines.

Now let's take a look at the tokenized part of the line. The parts of the program that get tokenized are the Basic commands, statements, and functions; the arithmetical, relational, and logical operators; and your variables. These are all keywords to the Basic interpreter. Numbers are also tokenized; we'll deal with them shortly.

Each Basic keyword is assigned either a one- or a two-byte token. One-byte tokens are used for all the Cassette Basic commands, statements, and variables, as well as the arithmetical, logical, and relational operators. One-byte tokens are also used for some functions. Most of the Cassette Basic functions, however, are represented by two-byte tokens, the first byte of which is FF (all token values in this article are stated in hexadecimal notation).

Keywords added to Cassette Basic by the two disk Basics are represented by two-byte tokens, the first of which is either FD or

FE. One special case is the ' command. When it's tokenized, it's converted to the tokens for : rem '.

Tables 1 through 4 set forth the Basic keyword tokens.

Now let's take a look at a sample program line. Consider this one:

Token	Keyword	Token	Keyword
81	END	B9	RANDOMIZE
82	FOR	BA	OPEN
83	NEXT	BB	CLOSE
84	DATA	BC	LOAD
85	INPUT	BD	MERGE
86	DIM	BE	SAVE
87	READ	BF	COLOR
88	LET	C0	CLS
89	GOTO	C1	MOTOR
8A	RUN	C2	BSAVE
8B	IF	C3	BLOAD
8C	RESTORE	C4	SOUND
8D	GOSUB	C5	BEEP
8E	RETURN	C6	PSET
8F	REM	C7	PRESET
90	STOP	C8	SCREEN
91	PRINT	C9	KEY
92	CLEAR	CA	LOCATE
93	LIST	CB	(unknown)
94	NEW	CC	TO
95	ON	CD	THEN
96	WAIT	CE	TAB(
97	DEF	CF	STEP
98	POKE	D0	USR
99	CONT	D1	FN
9A	(unknown)	D2	SPC(
9B	(unknown)	D3	NOT
9C	OUT	D4	ERL
9D	LPRINT	D5	ERR
9E	LLIST	D6	STRING$
9F	(unknown)	D7	USING
A0	WIDTH	D8	INSTR
A1	ELSE (if preceded by 3AH)	D9	'
A2	TRON	DA	VARPTR
A3	TROFF	DB	CSRLIN
A4	SWAP	DC	POINT
A5	ERASE	DD	OFF
A6	EDIT	DE	(INKEY$)
A7	ERROR	DF-E5	(unknown)
A8	RESUME	E6	>
A9	DELETE	E7	=
AA	AUTO	E8	<
AB	RENUM	E9	+
AC	DEFSTR	EA	−
AD	DEFINT	EB	*
AE	DEFSNG	EC	/
AF	DEFDBL	ED	^
B0	LINE	EE	AND
B1	WHILE	EF	OR
B2	WEND	F0	XOR
B3	CALL	F1	EQV
B4−B6	(unknown)	F2	IMP
B7	WRITE	F3	MOD
B8	OPTION	F4	\

Table 1. IBM pc one-byte Basic keyword tokens

```
100  INPUT A$:PRINT A$+" test"
```

The bytes used to store this line in memory are as follows:

88 0F	64 00	85	20	41 24	3A
address of next line	line number	token for INPUT	space	A$	:

91	20	41 24	E9	22	20	74	65	73	74	22	00
token	space	A$	token	"	Space	t	e	s	t	"	end of line

Notice that the two keywords in this line, *input* and *print*, have been replaced by their tokens, 85 and 91 respectively. Also notice that all characters that aren't part of keywords and that have ASCII codes in the range 20 through 7F are not tokenized; they are stored without modification. This is done for all variable names and identifiers as well as for text within quotation marks after a *print* statement or text following a *data* statement.

Tokens for Numeric Quantities. Basic also tokenizes all numeric quantities that appear in program lines. This is done in such a way as to minimize the number of bytes required to store the number in memory. Each token is typically followed by a certain number of bytes that contain the binary representation of the number.

The tokens used to indicate that a numeric quantity follows depend on the magnitude of the number and on the number type (integer, single-precision, or double-precision). If the number can be represented by an integer in the range of 0 through 9, then it's replaced by a single token with no data bytes following. If the number can be represented by an integer in the range 10 through 255, it is replaced by a token followed by one byte; the one byte following holds the number in binary form.

As the numbers get larger, the number of bytes required to store them increases. Numbers that can be represented by an integer in the range 256 through 32,767 are replaced by one token followed by two bytes. The two bytes hold the number in binary form (low-order byte first). Integers above 32,767, as well as all numbers containing fractional parts, are replaced by one token followed by four bytes (if single-precision can be used) or eight bytes (if double-pre-

Token	Keyword	Token	Keyword
81	LEFT$	93	STR$
82	RIGHT$	94	VAL
83	MID$	95	ASC
84	SGN	96	CHR$
85	INT	97	PEEK
86	ABS	98	SPACE$
87	SQR	99	OCT$
88	RND	9A	HEX$
89	SIN	9B	LPOS
8A	LOG	9C	CINT
8B	EXP	9D	CSNG
8C	COS	9E	CDBL
8D	TAN	9F	FIX
8E	ATN	A0	PEN
8F	FRE	A1	STICK
90	INP	A2	STRIG
91	POS	A3	EOF
92	LEN	A4	LOC
		A5	LOF

Table 2. IBM pc two-byte Basic keyword tokens (prefixed by FFH)

Token	Keyword	Token	Keyword
81	CVI	84	MKI$
82	CVS	85	MKS$
83	CVD	86	MKD$

Table 3. IBM pc two-byte Basic keyword tokens (prefixed by FDH)

Token	Keyword	Token	Keyword
81	FILES	8A	RESET
82	FIELD	8B	COMMON
83	SYSTEM	8C	CHAIN
84	NAME	8D	DATE$
85	LSET	8E	TIME$
86	RSET	8F	PAINT
87	KILL	90	COM
88	PUT	91	CIRCLE
89	GET	92	DRAW
		93	PLAY

Table 4. IBM pc two-byte Basic keyword tokens (prefixed by FEH)

cision must be used); the four or eight bytes contain the floating-point representations of the numbers (see "How Basic Stores Numbers," *Softalk*, June 1983).

Other tokens are used to indicate the number base in which the following bytes are to be displayed when the program is listed (hexadecimal or octal). The tokens used by pc Basic for this purpose are summarized in table 5.

The following examples should help clarify how the tokenization of numeric quantities is performed. Consider the following program lines:

```
(1)  100 X=5
(2)  100 X=10
(3)  100 X=256
(4)  100 X=32768
(5)  100 X=32768.13456
```

The tokens and data bytes used to represent the numbers following the equal signs are as follows:

```
(1)  16
(2)  0F 0A
(3)  1C 00 01
(4)  1D 00 00 00 90
(5)  1F 7F FA CF 9A 1F 00 00 90
```

The first byte in each sequence is the numeric token (see table 5), and the following bytes are the numbers stored in the format expected by the token.

One aspect of numeric tokenization deserves special mention. Some statements in program lines (such as *gotos* and *gosubs*) make specific references to line numbers within the program. When these lines are first typed in, the actual line number is tokenized and saved in memory as a two-byte integer—in the manner just described. However, after the program has been run, the tokens are changed so that it is the location of the line number in memory, not the line number itself, that's tokenized. As soon as the program is modified, however, these tokens are changed back to their original form.

The reason for this changing of tokens on the fly is to improve overall execution speed of the Basic program. On most other Basic interpreters, if you direct the program to another line by means of a *goto, gosub, resume, restore,* or *return,* the interpreter has to scan through the linked list of lines to find the one specified, and it must do this every time the statement is encountered. With pc Basic you only have to do this once, because after the first search the line-number tokens are changed to indicate the absolute addresses of the required lines.

Let's look at an example of how line-number tokens change on the fly. Consider the following trivial program:

```
100  GOTO 200
200 END
```

The tokens that are used to store this program begin at 0F72 if BasicA is in effect. Before you run this program, these tokens look like this:

7C 0F	64 00	89	20	0E
Address of next line	Line number (100)	Token for GOTO	Space	A line number follows

After the program has been run, however, the tokens become:

7C 0F	64 00	89	20	0D
Address of next line	Line number (100)	Token for GOTO	Space	The address of a line follows

7B 0F	00	82 0F	C8 00	81	00
Address of line 200 (−1)	End of line	Address of next line	Line number (200)	Token for END	End of line

You'll notice that the token preceding the two bytes reserved for the *goto's* line number changes from 0E, which indicates that the next two bytes are a line number, to 0D, which indicates the next two bytes are the address of a line number.

Elegant, isn't it?

Conclusion. The material presented in this article is based on personal research by the author. It's entirely possible that there are other tokens used by pc Basic that haven't been turned up by this research. Readers who find any are urged to communicate with *Softalk* so that we can all profit from the information. ▲

Token	Is followed by ...
0B	A two-byte integer, which is to be displayed in octal (&O prefix)
0C	A two-byte integer, which is to be displayed in hexadecimal (&H prefix)
0D	Two bytes, which are the address of a line number (−1) and not the line number itself
0E	Two bytes, which are the integer representation of a line number in the range 0...65529
0F	A one-byte, unsigned integer number in the range 10...255
10	(unknown)
11...1A	Nothing. These tokens refer to the single digits 0...9.
1B	(unknown)
1C	A two-byte, unsigned integer in the range 0...32767
1D	A 4-byte single-precision floating point number
1E	(unknown)
1F	An 8-byte double-precision floating point number

Table 5. IBM pc Basic tokenization of numerics

PASCAL from BEGIN to END.

by Bruce F. Webster and Deirdre A. Wendt

From time to time, you will find yourself wanting to be able to create and destroy data structures *while* a program is running. For example, we might define the following data type:

```
TYPE
  ship =
    RECORD
      sx,sy,qx,qy    : 1..8;
      energy         : 0..1023;
      shields,beams  : 0..255;
    END;
```

Now, let's suppose that the number of ships in existence during the course of a game would vary widely. What kind of data structure would we use to be sure we could handle all (reasonable) cases?

One solution inspired from our last column might be this:

```
VAR
  ships          : ARRAY[1..shipmax] OF ship;
  shipcount      : 0..shipmax;
```

where *shipmax* is a CONST value representing the maximum number of ships allowed during the game. The variable *shipcount* would be set initially to 0. Each time a new ship was created, the next free slot in *ships* would be used:

```
IF shipcount < shipmax THEN BEGIN
  shipcount := shipcount + 1;
  createship(ships[shipcount])
END
ELSE error('No more space for ships');
```

The procedure *createship* is one we would write. It would set all of the subfields of *ships[shipcount]* to the appropriate initial values. When a ship was destroyed, say *ships[indx]*, then we would "shuffle down" all the ships from *indx+1* to *shipcount*, like this:

```
FOR jndx := indx+1 TO shipcount DO
  ships[jndx-1] := ships[jndx];
shipcount := shipcount - 1;
```

This method is fine but has two drawbacks. First, we (the programmers) must decide ahead of time the maximum number of ships possible (*shipmax*). The program will never be able to handle any more ships than that. Second, space (memory) will always be allocated for the maximum number of ships, regardless of how many are actually in use. If we define *shipmax* to be 100, then we can never have more than 100 ships, and space for 100 ships will always be set aside, even if we never have more than one or two. Most of us are willing to live with these limitations—most of the time, they aren't much of a problem—but Pascal does offer a way around them: pointers.

Pointers. Let's modify our definition of *ships* as follows:

```
VAR
  ships          : ARRAY[1..shipmax] OF ∧ship;
  shipcount      : 0..shipmax;
```

If you look closely, you'll see that *ships* is no longer an array of *ship* but of ∧*ship*. The notation ∧<*data type*> refers to a *pointer* to the data type. In other words, *ships* is not longer 100 (or whatever) records but 100 pointers to records. "What's the difference?" you may ask. Simple. The record is some collection of data. The pointer is an address. If you still don't see the distinction, consider this: What's the difference between 100 people and the phone numbers of 100 people? With either one, you can talk to 100 people, but you need an auditorium for the former and only a phone booth for the latter. The analogy holds up in some areas and breaks down in others (space must still be allocated *somewhere*); but you can see the difference.

The next question is, how do we use a pointer? Or, better put, how do we use the record that the pointer points to? Answer: by pointing to it. If *ships* is an array of ∧ *ship* (that is, pointers to records of type *ship*), then *ships[1]* is a pointer and *ships[1]*∧ is the record to which it points. In other words, by sticking an up-arrow (∧) at the end of the pointer variable, we now refer to the data structure to which it points. For example, if we wanted to set the energy level of all currently allocated ships to 1023, we would do the following:

```
FOR indx := 1 TO shipcount DO
  ships[indx]∧.energy := 1023;
```

Note well that we cannot write *ships[indx].energy := 1023*, since *ships[indx]* is a pointer, not a record of type *ship*. The up-arrow makes a big difference.

The third question is, how do we point the pointer? In other words, how do we assign an address to the pointer, and how do we ensure that the address the pointer contains represents an area of memory that is not being used by anything else (operating system, program, other pointers, for example)? Answer: Pascal does it for us via the predeclared procedure *new*. For example, our ship initialization routine would look like this:

```
IF shipcount < shipmax THEN BEGIN
  shipcount := shipcount + 1;
  new(ships[shipcount]);
  createship(ships[shipcount]∧)
END
ELSE error ('Maximum number of ships allocated');
```

Note that the procedure *new* takes a pointer as its argument. It allocates the necessary amount of memory and sets the pointer to the appropriate address.

Linked Lists. We've partially solved our problem—we no longer have to allocate space for the maximum number of ships—but we still have to decide ahead of time what that maximum number is. And since that means that we always have to allocate space for that many pointers, we may not have gained that much of an advantage. Is there no way around this problem?

Obviously there is, or we wouldn't have written the lead paragraph as we did. The answer lies in creating what is known as a *linked list*. At this point, we face the temptation of referring you to Chapter 2 in the book *Fundamental Algorithms* (second edition) by Donald E. Knuth (Addison-Wesley; Reading, Massachusetts, 1973) and leaving it at that. However, having brought the subject up, we feel a certain obligation to tell you *something* about it. Here goes.

Let's modify our definition of the RECORD *ship* as follows:

```
TYPE

  shipptr            =∧ ship;
```

```
ship =
  RECORD
    sx,sy,qx,qy       : 1..8;
    energy            : 0..1023;
    shields,beams     : 0..255;
    next              : shipptr;
  END;
```

Those of you with sharp minds will instantly realize that we have violated the Great Underlying Rule of Pascal, namely that no identifier can be referenced until it has been declared. This, of course, is the exception to that rule. It's a necessary exception, since without it linked lists would be impossible (or, at least, quite messy).

Having modified our data structure, we now define our variables as follows:

```
VAR
  first,last        : shipptr;
  shipcount         : integer;
```

At the start of the game, we would initialize our variables as follows:

```
first := NIL;
last := NIL;
shipcount := 0;
```

(The predefined identifier NIL represents the value a pointer has when it's not pointing at anything.) When we wish to create a new ship, we call our procedure *addship*:

```
PROCEDURE addship;
BEGIN
  IF first = NIL THEN BEGIN
    new(first);
    last := first
  END
  ELSE BEGIN
    new(last^.next);
    last := last^.next
  END;
  last^.next := NIL;
  shipcount := shipcount + 1;
  createship(last^)
END; (* of PROC addship *)
```

(Note that we have a special test for creating a ship when none exists.) The pointer *first* always points to the first ship in the list. Each ship then points to the next one in the list via the subfield *next*. The pointer *last* always points to the last ship in the list. If there is only one ship in the list, then *first* and *last* both point to it. If there are no ships at all in the list (the list is empty), then both *first* and *last* equal NIL.

Now that we can create the list, how do we reference a particular ship in it? With the array it was easy: All we had to do was index into the array and presto! There it was. With the linked list, we've got to look for it. Here's a procedure to do just that. It takes the index value and returns a pointer to the appropriate ship (if it exists):

```
PROCEDURE fetchship(index : integer; VAR ptr : shipptr);
BEGIN
  ptr := first;
  WHILE (index > 0) AND (ptr < > NIL) DO BEGIN
    index := index - 1;
    ptr := ptr^.next
  END
END; (* of PROC fetchship *)
```

If index is less than or equal to 0, then *ptr* points to the first ship in the list. If *index* is greater than the number of ships in the list, then *ptr* gets set to NIL. Otherwise, *ptr* points to the appropriate ship.

Deallocation and Memory Management. At this point, we need to talk about how to reclaim the memory used by a pointer when you no longer need the data structure it points at. Standard Pascal provides the procedure *dispose* to take care of that. Going back to our array of pointers (*ships*), we could delete a particular ship this way:

```
IF ships[indx] < > NIL
  THEN dispose(ships[indx]);
```

This would set *ships[indx]* equal to NIL and make the memory that it had used available for other pointers. If we just wrote

```
ships[indx] := NIL;
```

then the memory would not be reclaimed.

Even when memory is reclaimed (via *dispose*), there arise the problems of memory management, fragmentation, and garbage collection (honest, that's what it's called). We don't want to get into all of the gory details (again, see Knuth), but here's a brief, simple description of the problem. Think of the heap (the memory used for dynamic allocation) as a long, narrow shelf. Each time we create a variable using *new*, we place a wooden block (width = amount of memory needed) somewhere on the shelf where it will fit. We, of course, have to decide somehow where to place it. That's memory management. Each time we destroy a variable using *dispose*, we remove the corresponding block from the shelf. After repeated calls to *new* and *dispose*, we can find that the free space on the shelf is tied up in lots of small, often useless chunks between blocks. That's fragmentation. To reclaim all of those little chunks, we shove all of the wooden blocks together and move them to one end of the shelf. Now, all of the free space is in one large chunk. That's garbage collection. It all sounds easy, but in practice it can be a real headache.

IBM Pascal supports the procedure *dispose*. However, we can find no documentation about how it handles the three problems mentioned previously. As such, we can make no assurances that everything will work well. *Caveat computor* (let the programmer beware).

UCSD Pascal does not support *dispose*. Instead, it uses a very simple memory management scheme. When you create variables using *new*, the blocks are placed next to each other, starting at the left end of the shelf (low memory). At any point, you can use the procedure *mark* to get the address where memory for the next variable would be allocated. You can then continue to create variables. Later, you can call the procedure *release* with the value you received from *mark*. Variables now created will be allocated starting at the address read when you called *mark*.

There are some important things to note here. Let's suppose that we executed the following code:

```
first := NIL; shipcount := 0;
mark(heapptr); (* NOTE: heapptr : ^ integer *)
FOR indx := 1 TO 10 DO
  addship;
release(heapptr);
```

We have created ten ships, then reset our next-free-location pointer (called the *heap pointer*) to where it was before those ten ships were created. This has done nothing to affect the linked list of ships. All the information is still there, and all of the pointers are still correct. The call to *release* has no effect until the next time we call *new*. At that time, the newly created data structure would start to overwrite the linked list. If we then read the linked list, funny things could result. The moral? If you use *mark* and *release*, be sure to set to NIL any pointers pointing into the freed-up memory.

Incidentally, UCSD Pascal provides two additional functions to help you with memory management. The function *memavail* returns the amount of free memory (in words, not bytes) available on the shelf. The function *sizeof*, given any data type or variable, returns the amount of memory (in bytes, not words) allocated for that variable or data type. With these four subprograms, you can do your own memory management and garbage collection. However, it can be a real pain.

Conclusion. There's a lot more we could say about pointers and linked lists, but we'll leave that for another column. Next month, we'll tackle files and file variables (which, by the way, are pointers). See you then. ▲

And it's happening again with our IBM® compatible Rana 2000. This 320K double density drive offers a large centering cone for problem-free diskettes and our exclusive silencing mechanism to make it the quietest disk drive you can buy.

At Rana we know the key to our success is providing the highest technology, on the best possible products, while filling the most possible user needs. That is why we spend so much time on research and development. Our world-renowned engineers were the first to offer increased capacity. The first to design a write protect feature. The first to use a metal band positioner and get 100% data integrity, and a 3 to 4 times improvement in access speed. And, the first to bring you all this performance, quality and dramatic styling.

This is why our Elite One got the #1 rating from Softalk Magazine. And our Atari® compatible Rana 1000 Slimline has turned a game computer into a sophisticated business tool. And soon we'll have a new Winchester drive and a new series of very high density minifloppies, for both IBM and Apple.®

So call or write for the nearest Rana retailer or computer store. We're Rana Systems. And we know that to keep a step ahead, we have put you first.

RanaSystems

21300 Superior Street, Chatsworth, CA 91311 213-709-5484
Call toll free: 1-800-421-2207. In California only call: 1-800-262-1221
Source Number: TCT-654

A Great Software

FriendlyWare PC Introductory Set

VOTED THE 5TH MOST POPULAR IBM SOFTWARE ON THE MARKET IN SOFTALK MAGAZINE'S RECENT READER SURVEY.

MONEY magazine described the Intro Set as one of the two best learner programs available for microcomputers.

PC magazine said it "had to be seen to be appreciated".

According to SOFTALK, "the implementation and sophistication of the Introductory Set are superior to anything available in the public domain".

The Intro Set is a must for novice users, but it's a lot more than just a starter set. Even seasoned pros are finding things they like among the 4 demo programs, 17 games, and 8 utilities on the package.

Some are even saying that several individual programs on the set (such as the Personal Check Book on disk #3) are worth the price of the package all by themselves.

SUGGESTED RETAIL $49.95

FriendlyWare PC Arcade

They told us that IBM'ers wouldn't buy games; but our customer feedback told us otherwise.

When the first two PC ARCADE production runs sold out before they shipped, we knew our customers were right.

PC ARCADE is a solid lineup of 10 fast-action arcade games for the IBM PC.

ASCII MAN, EAGLE LANDER, STAR FIGHTER TX-16, SHOOTER, BRICK BREAKER, GORILLA GORILLA, ROBOT WAR, BUG BLASTER, HOPPER and PC DERBY.

All 10 were written in Assembly language for speed and Text Mode for use on monochrome OR color monitor.

All games feature a PAUSE BUTTON and BOSS IS COMING KEY, and all but one of the games (EAGLE LANDER) can be controlled with EITHER a joystick or the keyboard.

And best of all, you can play them forever on five rolls of quarters.

SUGGESTED RETAIL $49.95

FriendlyWriter (with FriendlySpeller)

Combine the best little letter writer you've ever learned to use in 10 minutes, with a fast and accurate spell checker that compares your letter with a 30,000+ word dictionary, and you've got one of the greatest things to happen to word processing since the INSERT and DELETE keys.

FriendlyWriter is designed specifically for the kind of word processing you need the most – the 1 to 6 page business or personal letter. And using FriendlyWriter is as easy as striking a key.

FriendlyWriter (with FriendlySpeller), the only word processor in the world with a 30,000+ word dictionary that sells for under $100.

SUGGESTED RETAIL UNDER $70.00.

B Great Prices

The days of the $495.00 "Watzitdo?" software package are over.

The same is true for $39.95 one game packages.

Software pricing is coming down out of the clouds, and today's software consumers are proving once again that the old American business ethic of "producing a quality product at a fair price" still has merit.

At FriendlySoft, our objective has always been to give you the products you want, in a format that you can use and appreciate, at a price you can live with.

So far, your response is telling us that we're on the right track.

C THE BEST ~~Great~~ Support

No other software company in the world backs their products as completely and as simply as we do.

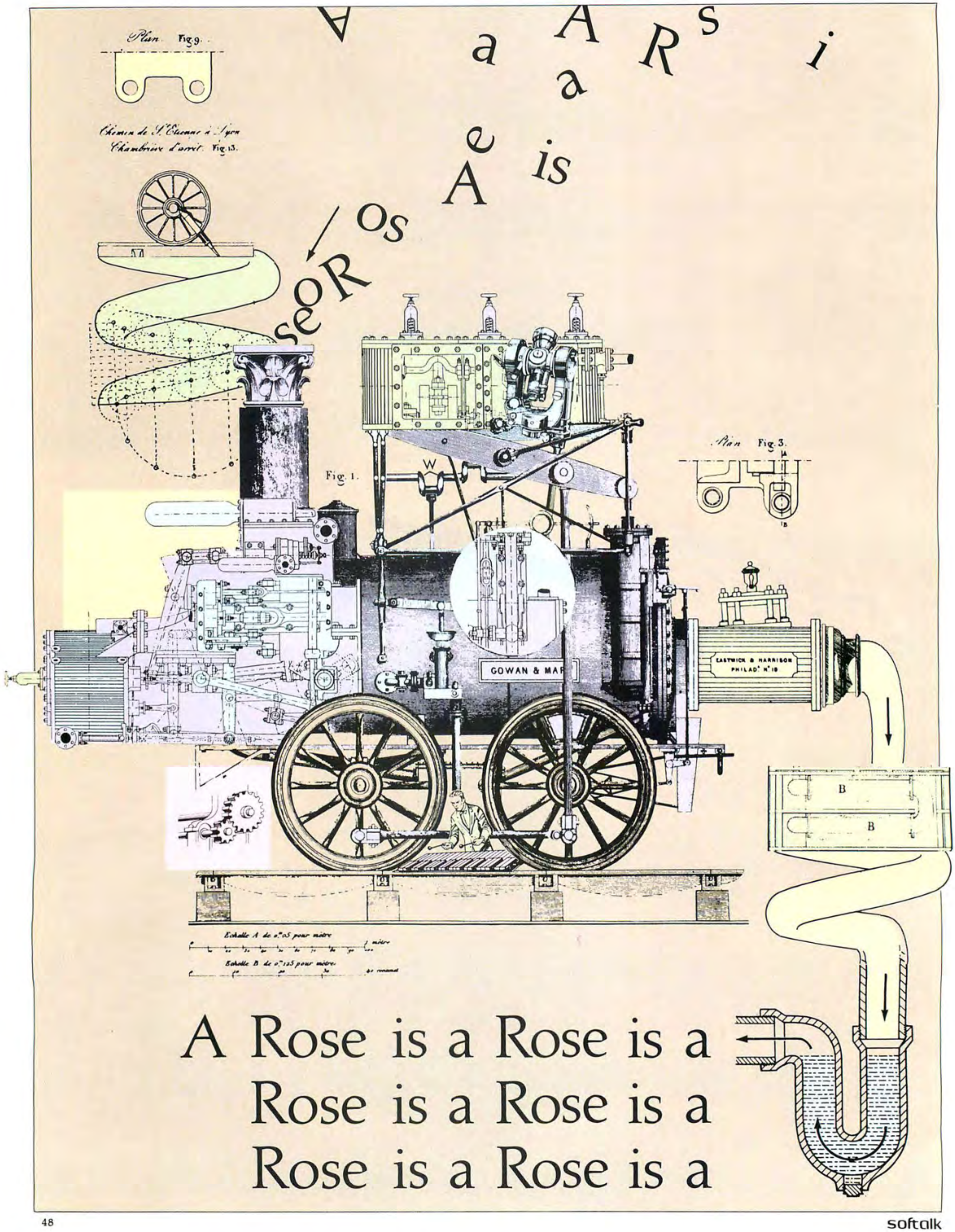
The FRIENDLYWARE "NO FINE PRINT" LIFETIME GUARANTEE states:

"If your master diskette fails due to normal use, we will replace it, AT NO COST, within 48 hours of receipt of your original master.

If your master diskette fails due to abnormal use (staple in media, food on hub ring) we will replace the programming onto your diskette within 48 hours of receipt of your original master and a blank, unformatted, diskette.

We also offer a unique 24 hour hotline, where you get quick answers, instead of taped messages.

In short, FriendlySoft is the unchallenged leader in software product support. And we intend to stay there.



Plan. Fig.9.
Fig.13.
Fig. 1.
W
GOWAN & MAR
EASTWICK & HARRISON
Plan Fig. 3.
B
B
A Rose is a Rose is a
Rose is a Rose is a
Rose is a Rose is a

COLUMN DEBUT

THE PROCESSED WORD

by Terry Tinsley Datz and F. Lloyd Datz

Welcome to "The Processed Word," a new column devoted to word processing on the pc and XT. In the coming months we'll review word processing programs and such related software as spelling and grammar checkers. We'll emphasize new and innovative programs of special interest, including VisiCorp's *VisiWord*, Bruce & James's *WordVision*, *PFS:Write*, *MultiMate*, *WordPerfect*, and the new entry from NBI, a manufacturer of dedicated word processors.

But we'll start by surveying the various features word processing programs may offer and defining some of the terms we'll be using later on. This column will be very general and elementary; if you're altogether new to word processing and haven't yet decided which program to use, you may find this installment a handy reference of features to consider when shopping for a program.

Overall Program Design

Menu-driven versus Command-driven Software. The overall organization of a word processing program is what determines how user-friendly it is. Menu-driven software lists all command choices on the screen, displaying more specific menus as general choices are made. In contrast, command-driven software provides no menus and only minimal prompts; consequently users must be familiar with all the commands. Most people find menu-driven programs easier to learn.

But the choice between menu- and command-driven software is not as obvious as it might seem. The helpful menus and prompts that make software so easy to learn also make it run considerably slower. Once you're familiar with most of your word processor's commands, you'll want to get the job done as quickly as possible without stopping to respond to screen messages or wait for a succession of menus to appear. Fortunately, some programs allow you to adjust the level of help to your particular needs. This means that all the menus and prompts are available to help you learn the program, and then as you become more proficient, you can turn off the prompts to speed the program up and leave more of the screen free for text display.

Mode. Another fundamental difference in the way word processing programs are organized has to do with modality. Programs that require you to make extensive mode changes are like cars with manual transmissions: You must keep them in the correct "gear" by entering and exiting the appropriate modes. For example, if you are typing along and then decide to go back and delete a few words, you have to shift out of insert mode and into delete mode, erase the unwanted words, shift out of delete, and go back into insert to continue typing.

Each program handles mode changes differently. Some automatically return to insert when you exit another mode, while others return to a "neutral gear" between mode changes. Many require mode changes for certain features but allow you to access others directly. Programs that keep mode changes to a minimum are, for obvious reasons, less awkward to use.

Text Entry and Editing Features

The Editing Screen. Most word processors display a status line at the top of the screen to provide you with such information as the name of the document in memory, the command currently being executed, the page, line, and column position of the cursor, the line spacing, and the current mode. In addition, there may be a "ruler" that indicates the settings for tab stops and left and right margins. A new trend, however, is to display fewer elements on-screen; this leaves the editing screen free of clutter.

Word Wrap. This standard word processing feature functions as an automatic carriage return. Instead of looking up at the end of each line to decide where to place the car-

Terry Tinsley Datz is a freelance writer and F. Lloyd Datz is a faculty member at the University of Utah Medical School. They have recently completed a book on word processing for the IBM pc.

riage return, you just keep typing, letting the computer decide where to end the line. When typing extends past the right margin, any words that won't fit on that line are automatically moved down to the left margin of the next line. This feature is a real timesaver, but it may be a while before you quit reaching for the return key when you near the end of a line.

On some occasions you'll need to control line breaks more precisely. Most, but not all, word processors allow you to turn off word wrap when you're typing highly formatted text, such as tables, forms, addresses, or programs.

Cursor Movement. When you generate text on a typewriter, the words originate where the typing element strikes the paper. On a word processor, text originates at the cursor position, which is usually indicated by a blinking line or square. Since entering or editing text requires that you first move the cursor to the desired location, cursor movement commands are the most frequently used editing commands.

The best word processors give you a variety of ways to manipulate the cursor. In addition to horizontal movement by character and vertical movement by line, most programs give you the option to move by the word, sentence, or paragraph; to the left or right margin; to the top or bottom of the screen; or to the beginning or end of the manuscript.

The trend in word processing programs is to allow you more and more flexibility in cursor movement. For example, most programs now allow you to jump to a specific page or character. This means, for example, that you can quickly add an entry to a resume by moving the cursor directly to the desired page number instead of having to browse through several pages until you find the page you need. Programs that won't jump to a specific page number sometimes permit you to place markers at strategic locations to which you may want to return later.

Scrolling. Scrolling allows you to review a document by scrolling the text up or down as if it were one continuous piece of paper. Many programs allow you to vary the scrolling speed, so you can slowly scrutinize the text for errors or quickly scan it to find a particular passage. Some word processors are capable of extensive horizontal scrolling; this is a big help when you're working with oversized documents, such as those used in accounting. If you routinely work with wide documents, look for a word processor that can horizontally scroll at least 132 columns. *WordStar*, the all-time champion of horizontal scrolling, goes far beyond the call of duty to scroll 32,000 columns!

Insertion and Deletion. To add text to a document you insert it at the position of the cursor. To prevent what has already been typed from being written over, text in front of the cursor is pushed forward as you enter the new words. If you're a fast typist, you might be able to enter text faster than it can be pushed forward. Therefore, some programs have a type-ahead buffer that acts like a scratch pad, saving any text entered too rapidly to be displayed. When you pause, the computer catches up, inserting the text it has stored. If the buffer fills up, a message is displayed asking you to wait a few seconds for it to empty.

Other programs split the text at the point of insertion and move everything in front of the cursor off the screen entirely. Although this approach prevents you from losing keystrokes, not being able to see the remainder of the sentence you're editing can be disconcerting.

When insertion mode is turned off, new text is typed over the characters under the cursor. Overstrike mode is useful for correcting typographical errors.

Deletion. While all word processors let you delete one character at a time, most also allow you to delete larger amounts of text—such as a word, a portion of a line, a screenful, or an entire document—by means of one or two keystrokes. Because deleting is so simple, you have to be careful not to erase text accidentally. Programs that automatically save deletions in a buffer graciously give you the opportunity to recover from such accidents.

Cut and Paste. What makes word processing such a powerful tool is the ability it gives you to rearrange sentences, paragraphs, and full pages of text with the press of a few keys. Changes that would normally require you to retype an entire manuscript can be accomplished in seconds.

Before you can manipulate a block of text, you have to mark the beginning and end of the passage. Then you can move the block, copy it to a second part of the document, or delete it. With some programs, block operations can be performed between two different files, so that text can be transferred from one document to another. The size limit for block operations is a function of both the word processing software and the memory capacity of the computer.

Search and Replace. Suppose you have a file containing several chapters of a book and you decide to edit Chapter 6. Instead of scrolling through the text until you find the chapter heading, you can use the search or find command, telling the computer to locate the heading "Chapter 6" and bring that portion of the book to the screen.

The search function can also be used to locate every occurrence of a word or phrase. Keep in mind, however, that the computer is literal to a fault: If you ask it to find all occurrences of the word he, the letters "he" will be found even if they are part of another

word. Some programs give you the option of circumventing this problem by marking the word only if it appears as a whole word (enclosed by spaces or punctuation). Capitalization can also cause problems. A word processor should be able to ignore case, if you wish, so that it will find both "computer" and "Computer" in response to the same command.

Related to search is a very important function called replace. Once a word or phrase has been searched for, replace lets you automatically exchange it for a different expression. In addition to doing a global replacement, some word processors will also replace the specified word or phrase only the first x number of times it appears or will show each occurrence to you in context and ask your permission to replace it. The ability to discriminate whole words is even more important with global replace than it is with global search. Without that capability, if you changed man to person, you would also, for example, change all occurrences of manipulate to personipulate.

The search and replace commands have a variety of uses. Form letters, contracts, or other documents with standard wording can be modified and used again and again, never losing their personal touch. Also, you can reduce typing time and errors by entering a symbol like "*" for lengthy or hard-to-type phrases. For example, when typing the manuscript for "2010: Odyssey Two," Arthur C. Clarke could have typed "*" every time he referred to Sivasubramanian Chandasegarampillai, the creator of the computer HAL. Then he could have used the global search and replace command to change every occurrence of "*" to the untypeable name, thereby saving himself considerable time and effort.

Columns. Arranging text or numbers in columns does not come easily to most word processors. As a result, programs vary significantly in their column-handling capability. Unfortunately, some do not recognize columns as independent portions of text. This means that any changes you make in one column will affect other columns that you have set up, thus destroying your carefully planned (and laboriously achieved) format. In short, if you frequently use columns or tables in your documents, look for a program that has a specific mode to handle them.

Some word processors that can work with columns go a step further to perform simple mathematical calculations, such as adding, subtracting, multiplying, and dividing across columns or generating subtotals, totals, and grand totals down columns.

Split-Screen Editing. Sometimes also called windowing, this feature enables you to view different parts of the same document or portions of two or more separate documents at the same time. In this way you can, for example, view the first draft of a manuscript in one window while you type the revision in the other. Although this feature is not a standard one, it is becoming more widely available. If you need windowing, look for a program that lets you scroll the text in either window regardless of which window the cursor is in. Having to move the cursor back and forth between windows each time you want to scroll just a few lines makes the windowing feature less efficient.

Formatting and Printing

A word processor's formatting and printing capabilities allow you to control the layout of your document. The best programs use on-screen text formatting, displaying the text exactly as it will appear on the printed page. An alternative is the use of nonprinting embedded commands to direct the printer to underline, right-justify, double-space, or perform numerous other formatting functions. In this case, the text appears on the screen unformatted, with the embedded commands scattered throughout. Many programs compromise between these two approaches, requiring embedded commands only for some of the more specialized features.

Not being able to see the final appearance of your document before printing makes it difficult to set up complex formats. Some programs will "print" to the screen first so that you can get a preview of what your document will look like on paper, but usually you have to get back into the editor to make any changes. Many users are satisfied with this approach and even prefer it, while others demand on-screen formatting.

General Layout. Text layout is controlled by means of a series of format settings. These are usually preset to commonly used default values, which can easily be changed. Some programs allow you to change the default values permanently; others require that they be reset each time you turn the computer on. A new trend is to store each document's settings on disk as part of the document file so that the format is ready to go whenever you work on that document.

Parameters commonly subject to format setting include the following: paper length, line height, pitch, line spacing, and tab stops.

Paper length is the total number of lines on a page, including top and bottom margins. Usually page length is set to sixty-six lines on a standard 8½ × 11-inch page.

Line height controls the vertical spacing of the text. It is usually stated as the number of lines per inch. A line height of six lines per inch is standard and compatible with a paper length of sixty-six lines per page.

Margins can be set just as they are on a typewriter. The top and bottom margins are the number of blank lines between the edge of the paper and the text. The left margin setting depends on the pitch (see following paragraph) and is the number of blank columns from the left edge of the paper to the text. With pica type, ten columns represent a one-inch margin. The right margin is set by entering the number of columns per line. A standard setting is sixty-five.

Pitch refers to the number of characters per inch. The most commonly used pitches are pica (ten characters per inch) and elite (twelve characters per inch). Unlike typewriters, word processors allow you to vary the pitch without changing the print element (assuming you're using a dot-matrix printer). This means that you can change the pitch in the middle of a manuscript without interrupting the printout.

Line spacing can also be varied as needed. Although manuscripts are usually either single- or double-spaced, many programs also allow you to specify more generous spacing between lines.

Tabs are commonly preset to intervals of five columns. Most programs let you change these settings to any column, making it easy to format outlines and indented text. In addition, if decimal tabs are offered, numbers can be aligned automatically by their decimal points.

Justification. The feature that sets text created on a word processor apart from typewritten text is full or right justification. Fully justified text looks like it has been typeset, because the right margins are perfectly straight. A word processor accomplishes this by inserting just enough extra space between words to align the last word on each line precisely at the right margin.

Inserting extra spaces between words can produce unevenly spaced text with large gaps between some words. Microspace justification, also called incremental spacing, improves the appearance of fully justified text by inserting tiny spaces (about 1/120th of an inch) between both words and letters. This evens out the spaces, giving the text a more pleasing appearance.

Proportional Spacing. Ordinarily, both typewriters and word processors set aside the same amount of space for each letter. Proportional spacing can be used to produce a more professional, typeset look by assigning different widths to different characters. This means that an "i" takes up less than half the space of a "w" on the printed page. If you produce camera-ready material or if your documents must have a very professional appearance, be sure that both the word processing program and the printer that you purchase can achieve proportional spacing.

Automatic Page Numbering. Most programs will print page numbers at the top or bottom of the page, either centered or at the left or right edge. If your documents are to

be bound, make sure that page numbers can be printed alternately at the left and right margins.

Headers, Footers, and Footnotes. A header is text that appears at the top of each page, prominently displaying such information as the name of a chapter or the title of a report. Footers, as the name implies, provide the same sort of information at the bottom of the page. Although many word processors automatically print specified headers and/or footers on each page, some programs limit them to a maximum of one line.

Footnotes present the formatter with a real challenge. The program must decide how much space to reserve at the bottom of the page, even though it doesn't know how long the footnotes will be or whether the writer will place an additional one just before the end of the page. Therefore, some programs do not even attempt to offer footnote capability. Programs that do support them vary as to how automatic the function is. The more powerful programs handle this problem beautifully, automatically numbering each footnote and moving it to the appropriate page when its referring text is moved. Be sure to check on length limitations; some programs set miserly limits, while others allow footnotes up to a page or more in length.

Pagination. Page breaks are usually indicated on-screen by means of a dashed line or other marker. This allows you to spot the inevitable first and last lines of paragraphs that are destined to be printed at the top or bottom of the page, isolated from the rest of the paragraph (these are often called orphans and widows, respectively). By adjusting the page breaks before you print your document, you can eliminate these unsightly creatures. Some programs automatically control orphans and widows by seeking out isolated lines and repositioning page breaks to eliminate them. A related feature allows you to set conditional page breaks to ensure that groups of lines which absolutely must stay on the same page (such as tables) do so.

Hyphenation. While some people feel that hyphenation is unnecessary and never use it, others are convinced that it is absolutely essential for creating attractive, fully justified text. If you routinely work with full justification, you will no doubt find that the extra spaces inserted to even out the margins make the words look too spread out unless some form of hyphenation is used.

Word processors handle hyphenation in various ways. A surprisingly large number ignore the problem altogether. Others allow you to insert ghost hyphens (also known as discretionary hyphens) at appropriate breaking points in long words; these word processors will then divide a word at the ghost hyphen, if the word falls at the end of a line. Another common approach is to show you words that are likely candidates for hyphenation—those destined to be wrapped down to the next line because of insufficient space on the current line. Usually the program places the cursor at a suggested break point but allows you to overrule its choice before it inserts the hyphen. The hyphen inserted in this manner will not print if further editing moves the word away from the right margin. A few programs hyphenate automatically, dividing words into syllables according to a set of built-in rules. This approach often leads to grammatically incorrect breaks, because of the large number of words that are exceptions to the rules.

Automatic Paragraph Reform. During the editing process, paragraphs may take on a disorderly appearance as you insert, delete, and otherwise rearrange text. Normally, in order to prevent your getting a printout with some lines dangling past the right margin and others consisting of only one word, you must remember to reform each paragraph individually as you finish it. Some of the newer programs perform this task automatically so that you can edit to your heart's content without worrying about cleaning up the mess afterward. These programs follow behind you, reformatting each paragraph according to the margin and line-space settings you've selected.

Special Printing Features. Provided that your printer cooperates, many word processing programs allow you to create special printing effects, including some not possible on a typewriter. Underlining (underscoring) is supported by most word processing programs, as is double strike; the latter produces a darker, solid-appearing print by having the printer strike each character twice. Boldface print can also be generated by having the printer type each character twice, with the second impression slightly to one side of the first. This produces a very dark, thick print, suitable for titles, headings, and key words.

Strikeout is the superimposition of dashes over characters. It's sometimes used to indicate deleted text in a revised version of a manuscript; it's also commonly used in legal documents. Overprinting is similar to strikeout, except that any character can be printed over another, making it possible to generate accent marks over letters in foreign languages and to combine several characters to form a symbol not found on the print wheel. Many word processing programs also enable you to create superscripts and subscripts, provided that your printer can handle them.

Special-character printing is done by sending a code to the printer so that characters present on the print wheel but not on the keyboard can be printed. In addition, on printers with ribbon-color selection, ribbon shift causes marked text to be printed in an alternate color, usually red.

File Handling

File Directory. Most word processing programs begin by displaying a directory of the files stored on disk. This directory often contains several pieces of useful information about each document, such as its name, its length in pages or bytes, the author's or operator's initials, and the date the document was created or last edited. Information about the length of the file is especially useful, since it gives you an estimate of how much disk space is still available. Some programs keep a tally of the amount of space used by all the files on each disk and tell you how much free space remains on each. This helps you avoid creating documents larger than the available disk space.

Ability To Save and Continue Editing. If the power going to the computer is interrupted or your equipment bombs out on you, the document in memory can be lost. Therefore interrupting your writing frequently to save your work on disk is a very good idea. Programs allowing you to execute a save without leaving the editor encourage you to save your work more often. On the other hand, those programs that make file-saving difficult and time-consuming will tempt you to court disaster by not saving your document often enough.

Automatic Backup. Some programs au-

tomatically save the last version of a file each time you begin a new revision. Then, if disaster strikes, your original text will not be lost. Such automatic backup, however, is not a complete substitute for saving your work frequently.

Automatic Disk Buffering. The maximum-size document that can be edited is related to one of two factors, depending on the word processing program. Some programs load the entire document into RAM; with these programs, file size is limited only by system memory. Other programs are capable of automatic disk buffering; in these, document length is limited only by the available disk space. Automatic buffering systems write and read portions of your document to and from disk as you edit. If you jump to a remote part of the document, the program stores the portion you're working on and loads the new section into memory. Although this means that you can edit lengthy documents in their entirety, it also retards program operation, since the program must occasionally ask you to wait while it goes to disk.

The newest approach is to keep only one page of text in memory at a time. This means that as soon as you finish a page, it is automatically stored on disk, safe and sound in the event of unforeseen disaster and readily accessed by page number for future editing.

Keyboard Macros. Keyboard macros are small segments of text, such as a standard closing for a business letter, that can be set up for quick insertion into documents. A collection of often-used phrases or sentences can be created and stored in a file; each of these phrases can then be assigned a specific key combination (such as control-A) or given a short name for identification. When you need one of the macros, you just press the appropriate key combination or enter its name, and the text is written to the screen almost instantly. Macros can also be used to make format changing quick and easy.

Document Assembly. The most powerful word processors allow you to put documents together with the help of various files that you've created and stored on disk previously. This file-merging process (sometimes called boilerplating) can be automated to help you create personalized letters from a mailing list or assemble several standard paragraphs to produce a final document, such as a contract or a will. Document-assembly capability varies widely among word processors; some require you to interface a second program with the basic word processor.

Printer Control. A word processing program's file handler usually gives you several options for controlling the printing process. A document can be printed one page at a time using single-sheet feed or a continuous feed if the paper is tractor-fed. Some programs also allow you to print multiple copies of the same document with a single command.

If your computer has adequate memory, background printing may be possible. Normally, when a document is being printed, the computer cannot be used until the printing is completed. Background printing frees a portion of the computer's memory so that you can edit one document while printing another. If you produce many long documents or if rapid throughput is important to you, you'll appreciate this feature.

Batch printing can also be used to save printing time. Programs that support this feature allow you to specify a list of documents to be printed; this list is sometimes called a print queue. The word processor automatically supervises the printing process, making as many copies of each document as you need. Since printing can be done without intervention, you're free to work on other projects.

In Coming Months. Next month we'll get down to business and start reviewing word processors available for the pc. First on the schedule are two programs from Satellite Software International, *WordPerfect* and its baby brother, *Personal WordPerfect*. ▲

DOSEASE™

The MENU Interface to your IBM PC-DOS 2.0

Are you tired of forgetting the PC-DOS commands that you need? Do you wish you knew all the other functions that this powerful system can provide? Do you wish there was an easy way to use the right command at the right time? DOSEASE is the SOLUTION.

DOSEASE makes it EASY

DOSEASE brings you the power of PC-DOS at your fingertips. The main menu gives you a list of all PC-DOS commands. YOU now have access to ALL the system functions without having to remember ANY cryptic commands.

DOSEASE displays a menu that describes the function you selected and gives you the available options. You select the desired options and enter the required data which DOSEASE verifies.

Press the ENTER key when you have made your selections. DOSEASE then commands PC-DOS to perform the requested function. When this function has been completed, DOSEASE returns a menu so you can continue your work.

DOSEASE

FILE	DISK	DIRECTORY
10. List Directory	20. Format disks	30. Change directory
11. Type a file	21. Check disk	31. Make a new directory
12. Copy files	22. Copy disks	32. Remove a directory
13. Erase files	23. Compare disks	33. List directory tree
14. Print files	24. Assign drive	34. Set directories that have execution files
15. Compare files	25. Reset assignments	
16. Combine files	26. Transfer DOS files	
17. Rename files	27. Set Verify on Write	
18. Recover bad files		
SYSTEM	**FIXED DISK**	**OTHER**
40. Change Date	50. Back up fixed disk	60. Set Graphics Print
41. Change Time	51. Restore fixed disk	61. Run Batch
		62. Set Program Break

Please enter the command number: ___

BACKUP

Backs up one or more files from a fixed disk drive to a diskette drive.

Source file information:
Drive: _____ Filename: _____ Ext.: _____
Path: _____

Target drive: ___

Date (if files written on or after this date are to be backed up): __/__/__
All sub-directories below the specified path are to be backed up (y/n): ___
Only modified files to be backed up (y/n): ___
Is the diskette in the target drive the backup diskette (y/n): ___

DOSEASE makes it FAST

DOSEASE lets you interact with PC-DOS with an average of 30% fewer keystrokes. You can now get your job done faster.

DOSEASE adds COLOR

DOSEASE makes judicious use of the highlighting and color capabilities of your screen to simplify your interaction with PC-DOS. You no longer need to be resigned to the drudgery of the PC-DOS environment.

DOSEASE pays DIVIDENDS

DOSEASE can substantially increase the return on your significant investment in IBM-PC or IBM-PC XT with PC-DOS 2.0. DOSEASE is available at a very affordable price of $60.

Software Solutions, Inc.

305 Bic Drive, Milford, CT 06460 • (203) 877-9268

For information, please write or call. Dealer inquiries welcome.

Photography by Tim Durr

COLUMN DEBUT

MICRO FINANCE

by Ken Landis

If you've ever made an investment decision, you know there can be only three kinds of return on your investment: positive, zero, or negative. Given this three-state universe, we might assume that picking the decision that gives us a positive return has a probability of 33.3 percent. We *might* assume that, but we'd be wrong. With literally millions of potential investments, the probability of picking a profitable one is much smaller than 33.3 percent.

Improving the Money-Making Odds. So the question is, "How do I improve my odds of making money?" That's the question we'll address every month in this column. "Micro Finance" will show you how to solve financial questions—whether personal, corporate, or public-sector—using your pc as a tool.

Too much emphasis has been placed on reviews of software. Reviews are fine, because they give potential software buyers an indication of what to expect if they buy certain packages. But they don't show *how* to use that software; that's an area we'll address in this column.

The IBM pc didn't start the microcomputer revolution, but in its short history it has already greatly enlarged the market for personal computers. In the corporate and public sectors, it has "legitimized" the role of the personal computer. Management Information Systems (MIS) directors who may once have thought that personal computers were a flash in the pan now recognize that they're here to stay and that they play an important role in any organization's data processing.

On the personal front, the attention that the IBM pc has received has expanded the microcomputer marketplace. Not all home computer users buy pcs (as you can tell by looking at the hundred or so home computers available), but once a family has tasted personal computer power, it's not likely to keep living and working the way it used to.

To use an old expression, the IBM Personal Computer is the Cadillac of the personal computer business. Interestingly enough, though, we have more than one model of Cadillac. It would be very foolish of us to ignore the pc look-alikes. A majority of them are highly compatible with pc hardware and software. This group of products represents a large user base of MS-DOS software and IBM pc peripherals. For users of these other systems, this column will attempt to highlight any procedural or functional differences in the products.

The software systems we'll be discussing, using, evaluating, and generally running through the mill can be categorized as follows:

Spreadsheet software—software such as Lotus's *1-2-3*, Context's *MBA*, *Multiplan*, *VisiCalc*, and others. The primary characteristic of these packages is that they're electronic blackboards, capable of mathematic manipulations. Some of the systems have data-handling, graphics-generation, and/or communications capabilities as well.

Financial modeling languages—such as *DSS/F*, *Financial Planner*, and *Micro FCS*. These software systems use English-like procedural commands to build financial models. They are based on mainframe-developed decision support systems (DSSs) and are powerful (and sometimes expensive) tools for solving complex problems.

Dedicated software packages—such as real estate analyzers, tax compliance and planning systems, and industry-specific financial analysis packages.

Investment software—packages for analyzing technical or functional securities indicators, accounting for portfolio trans-

Ken Landis is a microcomputer and office automation specialist with the Saint Louis office of Peat Marwick.

actions, or retrieving financial information from remote databases.

Utilities—programs designed to work in conjunction with analysis software. These packages include graphics-generation systems, financial programming utilities, templates for use with spreadsheet programs, and data communication packages. These software products, when used in conjunction with the other classes of software, help provide a pc-based solution to problems.

Database management systems (DBMSs). These packages are not customarily thought of as "financial" in nature, but when properly customized they have tremendous analytical power.

There's a basic principle that must be understood from the start: A pc will not solve all your problems. It can and will help you with only those problems that naturally lend themselves to computer-based solutions. It cannot take the place of a mainframe computer system. It won't get your boss to like you if he doesn't care for the color of your tie. And it will not solve your home budget problems. What it can do is help you make better decisions by providing you with more accurate information, faster and more efficiently than you could get it manually. It seems that a lot of pc users don't understand this. If we encounter a problem that is better solved manually, we'll tell you. But if a pc fits the bill (as it does a great deal of the time), we'll tell you that as well and show you how to use it to solve the problem.

The Necessary Ingredients. For a majority of the programs dealt with in this column, you'll need an IBM pc with at least 128K of RAM. Some programs use more memory, some less, but 128K is getting to be the magic number. You'd do well to have at least two double-sided floppy drives (or a floppy and a hard disk) and a printer capable of at least 132 columns in its compressed mode.

Options that will be required for some of the programs include a color/graphics adapter, a color monitor, a modem, graphics firmware for your printer, a plotter, and subscriptions to remote database services, such as Dow Jones News/Retrieval, CompuServe, Warner, or the Source.

Very few of us have all we'll need for every program. But experience indicates that nine times out of ten you already have what you need for the applications you're planning to run. If you don't, then "Micro Finance" will help you decide whether you really need to make an additional purchase. In many ways, this column will be an unbiased, professional advisor—telling you only what you need, not what a salesman wants you to think you need. In this capacity, we'll be happy to try to answer readers' questions. We can't provide investment or business advice, but we can help you structure your solution to fit your application.

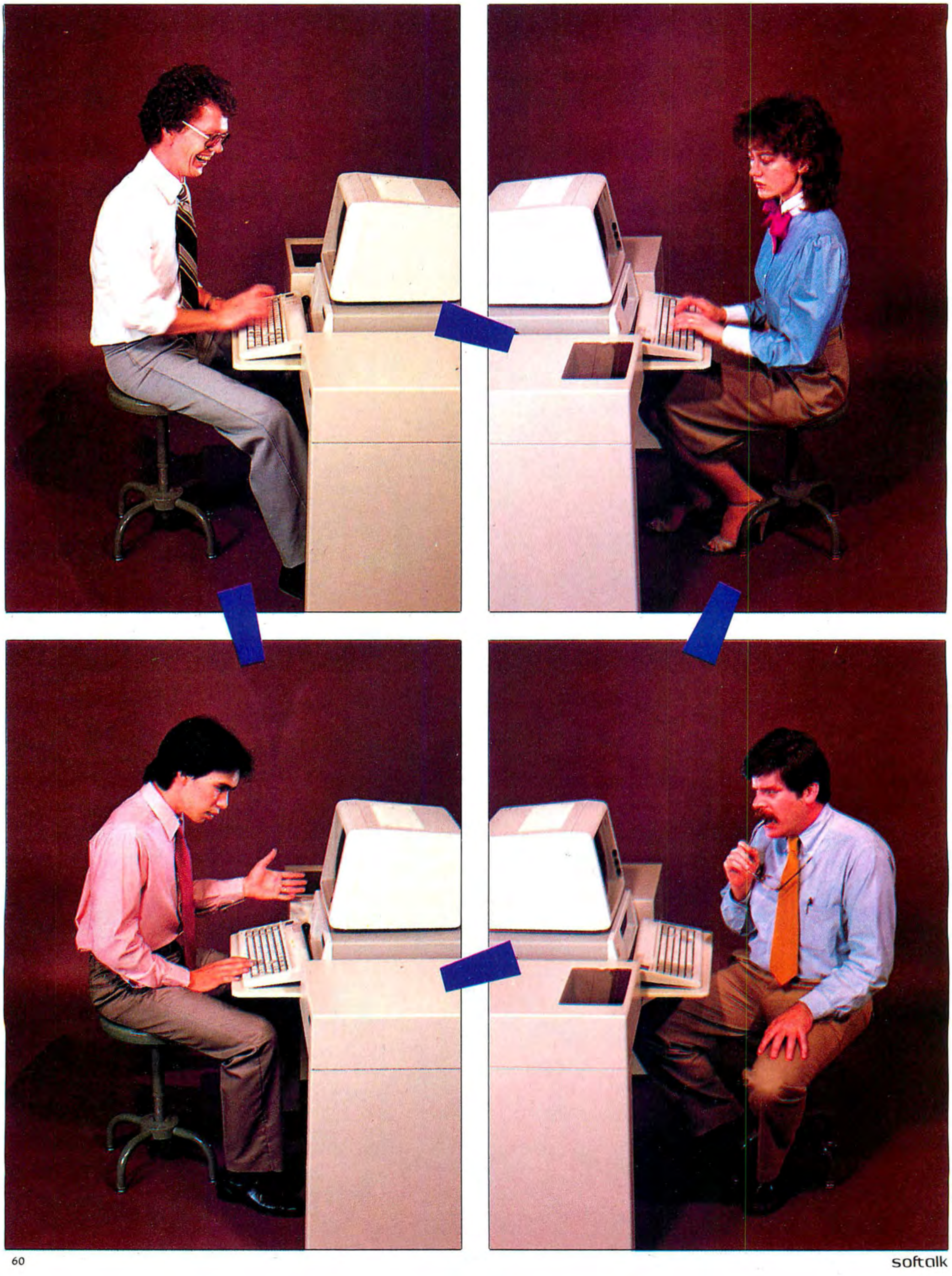
Your pc can help you only as much as you help yourself. By gaining a basic understanding of what your computer is and isn't capable of, you'll be in a better position to use it well. We also intend to help you learn, or brush up on, the financial knowledge you need to solve a wide variety of problems.

In "Micro Finance," we'll not only look at software and hardware, we'll also look at books, training aids, and other information resources that can help solve problems.

Your comments, suggestions, and experiences are more than welcome. A reader forum for the sharing of information will be featured each month. A powerful software package or information resource might slip through the cracks because it's new or has only regional distribution. If that happens and you know about it, share this information with us and we'll get right on it.

Next month: Should I buy or lease my home, and what happens if I decide to move? What are the income effects I should look at, and how do I decide which alternative is best?

See you then. ▲



Part Two

A PRIMER ON NETWORKING

The following is the second in a series of articles excerpted from the book Communications and Networking for the IBM PC, *by Larry Jordan and Bruce Churchill, and published by the Robert J. Brady Co.*

The type of network a business or organization selects will have a major bearing both on what the network costs and what it will do. Therefore, anyone planning to become involved in the choosing process would do well to become as knowledgeable as possible about the various network configurations and to weigh the strengths and weaknesses inherent in each of them.

With that in mind, we'll look this month at how local area networks are classified and at how they vary in terms of cost, capability, flexibility, growth potential, and capacity.

Networks are usually classified according to three characteristics: bandwidth, topology, and protocol. Bandwidth has to do with a network's data path capacity, topology simply means the network's physical structure, and protocol is the set of rules by which a network controls access by its devices.

by Bruce Churchill

Bandwidth. Bandwidth is the measure of a network's ability to move data. Considered from this perspective, there are two kinds of local area network systems—baseband systems and broadband systems.

The Xerox/Intel/DEC Ethernet, the Corvus Omninet, and the Orchid Technology PCNet are all examples of baseband systems. In the baseband system illustrated in figure 1, a single digital signal—a serial stream of bits tightly arranged in formatted data packets—is input to the network. These serial data packets are sent and received at a specific design data rate, typically one to fifty megabits per second. (By way of comparison, the ordinary telephone system, not including those segments that have been upgraded for digital transmission, can support a data rate of about 120 to 240 characters per second with reasonable accuracy.)

In a baseband system, data transmission is completely digital and operates at half duplex. In other words, a node (that is, a workstation on the network) can either send or receive but cannot do both at the same time.

The local area networks we'll discuss later on in this series are baseband networks. Baseband networks owe their popularity to their ease of installation and relatively low cost.

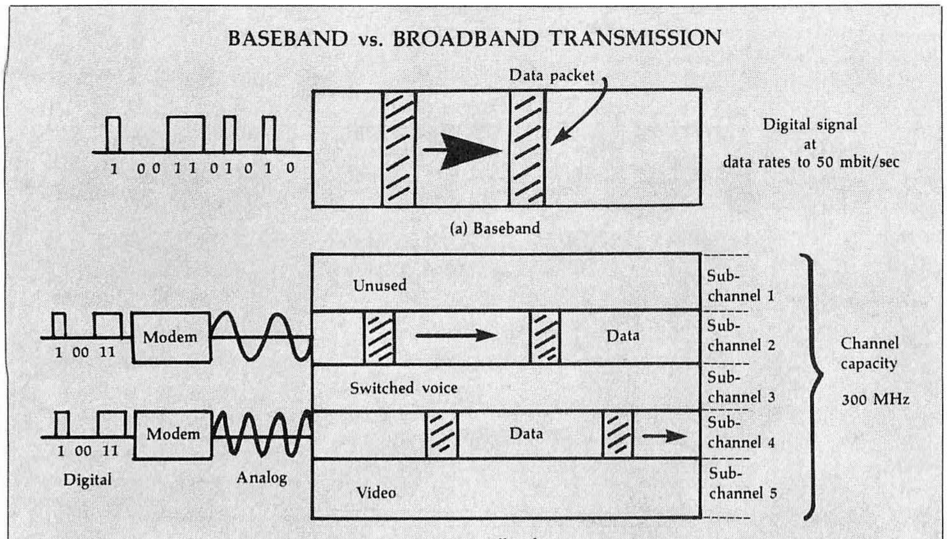


Figure 1.

A broadband system, such as the one illustrated in figure 1, sends data along high-capacity cable television (CATV) distribution media. This type of cable has a bandwidth of 300 megahertz (MHz). The high bandwidths on CATV cables used in broadband systems were originally designed to accommodate fifty channels of 6 MHz television video for community distribution. This capacity can be fully used in broadband local area networks. Wang Laboratory's Wangnet and Sytek's Localnet 20 are two examples of broadband systems.

With broadband systems the bandwidth (which is also called channel capacity) is normally partitioned into subchannels, each of which is assigned a specific purpose or function. Examples of functions would include high-speed data transmission, low-speed data transmission, video, or switched voice. This technique of partitioning the bandwidth is known as frequency division multiplexing, or FDM. FDM allows the organization using the network to select subchannel bandwidths according to the service desired in each channel. Figure 2 provides one example of how the subchannels on a broadband LAN might be allocated.

In a broadband local area network, data packets are transmitted as analog signals, just as they are on long-distance modem networks; therefore broadband LANs require modems for digital-to-analog conversion. A high-speed data channel also requires a high-speed (and high-cost) modem.

The added capacity and diversity of service that broadband LANs provide come at a price—namely, a substantial increase in cost and complexity. The requirement for relatively expensive variable-frequency and high-speed modems, along with the increased complexity of the communications interface units (modem adapter cards), drives up costs. In addition, installation of broadband LANs is difficult. The placement of components is critical, and careful tuning is required for each network on a subchannel. For these reasons, it's not likely that broadband technology will find its way into low-cost personal computer LANs.

Topology. Topology, a fancy word for a simple concept, refers to the way networks are physically connected together. The three common LAN topologies—star, ring, and distributed bus—are illustrated in figures 3 through 5. As you might expect, there are variations and combinations of these configurations, but we'll limit our discussion to the types shown here.

A desirable attribute of any local area network is the absence of a critical node. A critical node is any node whose failure will cause the entire system to fail. The worst topology, from this standpoint, is the star network, illustrated in figure 3, in which the central computer is a critical node. In the ring topology, shown in figure 4, critical nodes may or may not exist, depending on the network in question. (IBM's local area network for the pc has not been announced as of this writing, but it will in all likelihood be a ring topology.) The distributed bus topology depicted in figure 5 (and featured in last month's figure 1) does not have a critical node.

The key features that managers should be aware of in selecting a network topology are reliability and ease of expansion. The inherent reliability of distributed bus networks and their flexibility in accepting new devices make them a logical choice for low-cost networking applications. All of the IBM

BROADBAND CHANNEL ALLOCATION.

48 full duplex point-point channels with 9.6 Kbits/sec each	300 MHz bandwidth
32 full duplex point-point channels with 64 Kbits/sec each	
128 switched point-point channels with 9.8 Kbits/sec each	
1 high-speed data channel at 10 Mbits/sec	
5 standard CATV video channels at 6 MHz each	

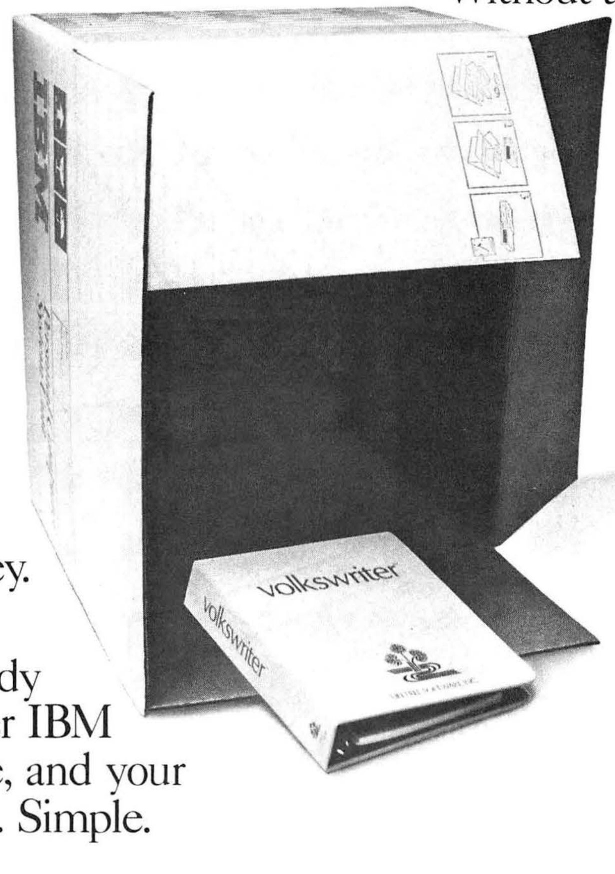
Figure 2.

Volkswriter makes word processing so simple, you may forget to read the instructions.

You learn to use Volkswriter right on the screen of your IBM Personal Computer or compatible. The lesson is in plain English. And in minutes, you're a pro. Simple.

When you need extra help, you don't even have to ask. Volkswriter tells you which key to press to perform your chosen task. One task, one key. Simple.

Volkswriter is ready to use with 1-2-3, other IBM or compatible software, and your own. You just slip it in. Simple.



Volkswriter lets you use your choice of fifteen top printers. Without any reprogramming. Simple.

When you buy Volkswriter for $195, you're getting word processing that's faster, more responsive and simpler to use than others selling for over twice the price. Smart.

And in case you're still wondering which word processor is your wisest choice, the answer is simple.

volkswriter™

The word processor for your IBM Personal Computer or compatible

LIFETREE SOFTWARE, INC.
brings the world within your reach.

411 Pacific St., Suite 315
Monterey, CA 93940
(408) 373-4718

pc networks we'll cover later in this series are of the distributed bus topology.

Protocol. A protocol can be thought of as the set of rules by which data communications are conducted. Before we can say more about specific LAN protocols, we need to digress for a moment to discuss the ISO seven-layer data communications architecture.

The ISO (International Standards Organization) is a worldwide group composed of standards organization representatives from member nations. The American National Standards Institute (ANSI) represents the United States. The ISO develops international standards for data communications.

A seven-layer model was developed by this organization to define a universal architecture for interconnecting heterogeneous computer systems. Although this architecture was developed for large computer networks, its rules apply to the IBM Personal Computer as well when the pc's communications extend to remote computer systems or to local area networks.

As a user of a local area network, you'll seldom need to concern yourself with the details of this seven-layer architecture. In fact, the existence of the layered design actually aids in hiding the complexity of communications from you the user. Nevertheless, since you're bound to see these terms thrown about in what you read about networking, it's worth having an overview of the ISO model.

Here is a brief description of each of the seven layers:

Layer 1: Physical Control. This layer in-

LOCAL AREA NETWORK TOPOLOGIES

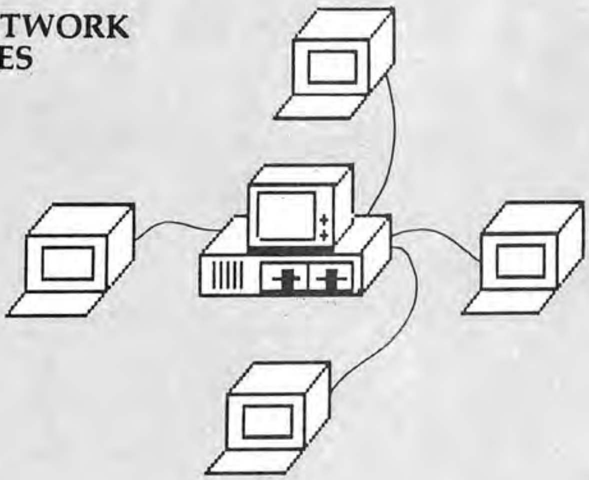


Figure 3.
Star network

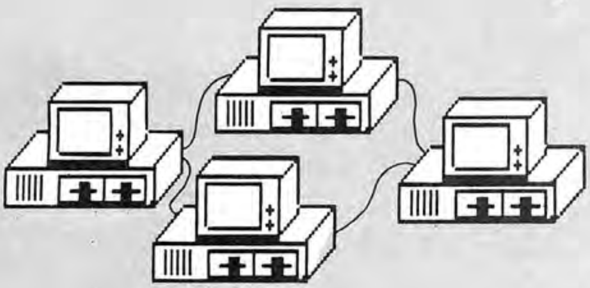


Figure 4.
Ring network

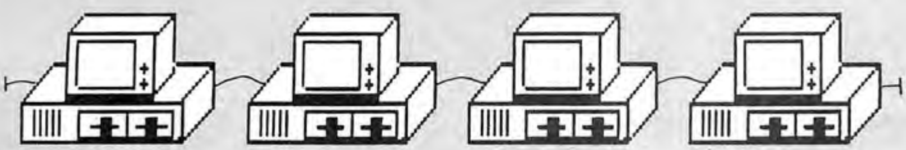
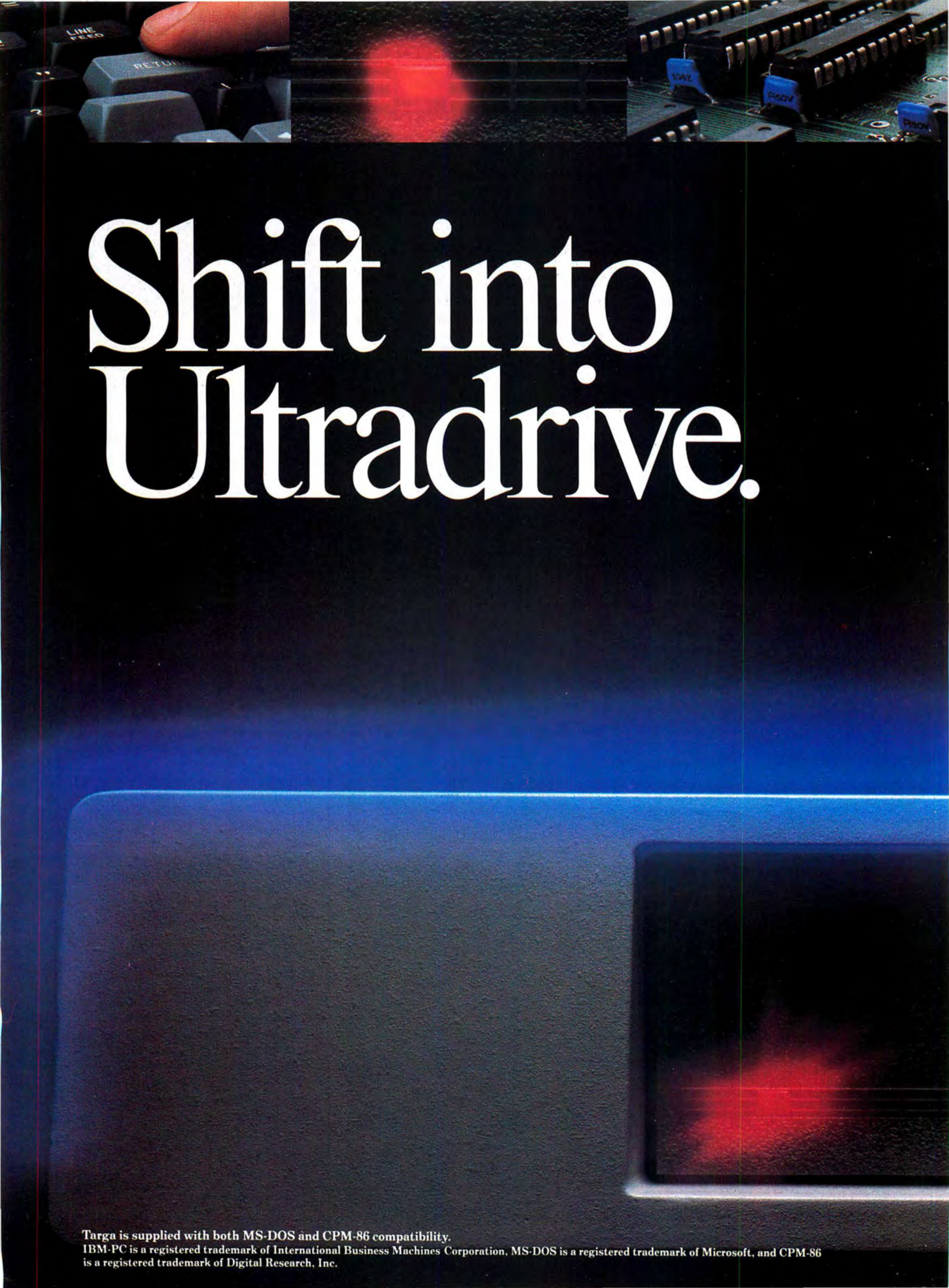
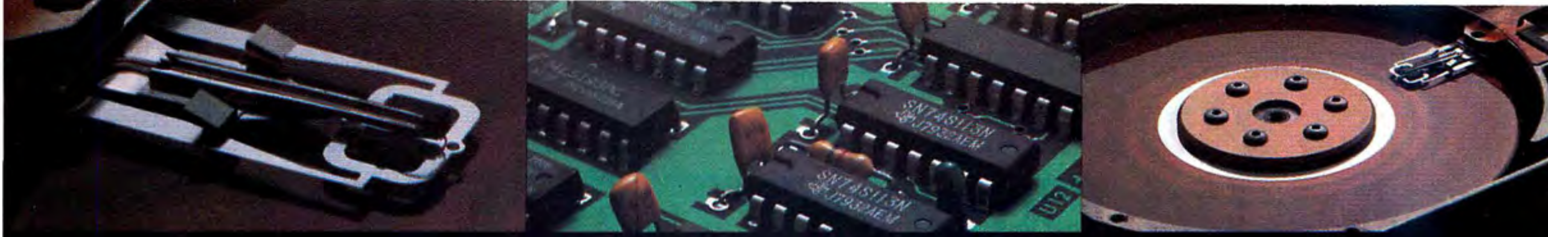


Figure 5. Bus network



Shift into Ultradrive.
Targa is supplied with both MS-DOS and CPM-86 compatibility.
IBM-PC is a registered trademark of International Business Machines Corporation, MS-DOS is a registered trademark of Microsoft, and CPM-86 is a registered trademark of Digital Research, Inc.



Targa®: The ultimate mass storage machine for your IBM®-PC.

With up to 30 megabytes of hard disk capacity, you can handle data bases that would fill phonebooks. And access, process, and update that data in seconds.

Which, after all, is what serious personal computing is all about.

The Purr of Performance.

Targa is one beautiful piece of engineering.

Fast. Data glides in and out at 5 million bits per second.

Quiet. Even with disks spinning at 3600 rpm inside.

And solid. Which is why Targa carries a warranty few others would dare to offer. 24 full months.

Automatic Is Standard.

Just plug the Targa into your PC, and you're ready to go.

Targa comes with plain English instructions and plenty of software. Easy to use, clearly documented programs help you transfer data in and out, use powerful system utilities, and diagnose problems.

The Luxury Model At Economy Prices.

The best way to order a Targa is right from the CMC factory in Seattle, one of the world's oldest microcomputer hard disk manufacturers. You pay considerably less than you would for a lesser system sold at a computer store.

Powerful standalone Targa mass storage systems start at $1550.

Targa 50	5 megabyte models from $1550
Targa 100	10 megabyte models from $1890
Targa 150	15 megabyte models from $2190

Prices include shipping and handling. Call for prices on expanded capacity models.

Take A Test Drive.

No risk.

We'll ship your Targa factory direct. Then give you 30 days to make sure you're satisfied. If not, we'll refund your full purchase price. Plus your return postage costs.

To reach our professional sales staff, call toll free, **800-426-2963.** (In Washington, dial **206-885-1600.**)

You can charge the Targa to any major credit card. Or you can send a check, money order, or wire transfer. Your Targa will arrive within days.

And your IBM-PC will get a much needed boost in mass storage horsepower.

cludes rules that apply to the physical hardware used to transfer data bits. Interface cables and connectors covered by the EIA RS-232-C and RS-449 standards fall into this category.

Layer 2: Link Control. Layer 2 includes rules that apply to the transfer of blocks of data over a physical link. This layer governs the information that must precede and follow blocks of data and defines a protocol for the data transfers. The synchronous data link control (SDLC) used by some mainframe networks falls into this category.

Layer 3: Network Control. This layer provides rules that apply to computers that are communicating but not connected to the same physical line. This layer governs the use of multiple lines and circuits used during communications and ensures that discrete packets of information are delivered to remote computers in the same sequence as they were transmitted.

Layer 4: Transport End-to-End Control. This governs the end-to-end integrity of transmission between two users, to prevent lost transactions, prevent double processing of transactions, control the flow of transactions, and ensure proper addressing of user machines. This layer provides a standard user interface with a transport service, regardless of the type of network used.

Layer 5: Session Control. This layer governs the process of setting up or terminating a communication session. It checks to determine if proper communication is taking place. If proper communication is not taking place, this layer must restore the session without data loss or, if that is not possible, terminate the session in accordance with specified rules.

Layer 6: Presentation Layer. Layer 6 governs the character set and data code used for communication. Printer and screen displays are also governed by this layer. The conversion of one character set to another and the compacting of a character stream into a smaller bit stream are also controlled by this layer. Communication software design that involves character code transmission and terminal emulation falls into this category.

Layer 7: Process Control. This governs applications of communications that interface with other high-level functions, such as distributed database activities and file transfers. This layer prevents data transfer integrity problems or data transfer speed mismatches with receiving devices.

Ultimately, the significance of a layered communications model is economic: Without some standardization of protocol within layers or interfaces between layers, no two communications systems would be compatible. The aim is to establish a local-area-network specification with recognized interface and protocol standards that will al-

Table 1. Summary of LAN features—an overview.

Category	Variable	Least Cost	Most Flexibility	Most Growth	Most Capacity
Bandwidth	Baseband	X	X		
	Broadband			X	X
Topology	Star	X			
	Ring				
	Bus		X	X	X
Protocol	Polling	X			
	Token Pass			X	X
	CSMA		X	X	

low networks and devices from different vendors to communicate with each other. This may, of course, be an unattainable goal in practice, but if a real effort is made, the proliferation of hardware and software can at least be held to reasonable levels.

The best effort to date in this regard is the work being done to develop the IEEE local net specification. Since no agreement could be reached on a single specification, IEEE 802 is evolving into a dual specification for distributed bus systems (such as Ethernet's) and ring systems (such as IBM's).

To get an idea of the importance of standard protocols, consider the case of a business person who has purchased a low-cost local area network advertised as "Ethernet-compatible." Saying that a network is Ethernet-compatible means that the protocols for layers 1 and 2 of that network follow the Ethernet specification. Therefore, if the same individual wanted to expand the operation into a full-blown Ethernet later on, his existing system would still be usable in that larger LAN. The operating software might have changed over time, but the physical components and the protocol implemented on the network adapter card would not.

A manager who seeks to choose the best LAN for his or her business needs will be most interested in the protocol that governs access to the network. This protocol is found in both the physical and the data-link layers. The most common such protocols are polling, token passing, and carrier sense multiple access (CSMA), which are found respectively in star, ring, and distributed bus networks.

CSMA protocols can also include collision-detect features; in such cases, the complete protocol is referred to as CSMA/CD. The emphasis in these articles is on the CSMA technique, because the IBM pc networks currently available are of the distributed bus type.

The presence of higher protocols (layers 3 through 7) in a local network design varies depending on the particular network. Omninet (to be described in a subsequent installment) implements layers 1 through 4, while Ethernet specifications define layers 1 and 2. As the user base of LANs grows, the interfacing of application programs to local networks will be of even greater interest. Thus, it is likely that layers 6 and 7, the presentation and process control (applications) layers, will receive increasing attention in future LAN design. Layer 6 in particular is important to the communications of graphics images.

Summary of Local Area Networks—Choice Factors. Earlier in this discussion of local area network types, we identified cost versus capability as important factors to consider when choosing a local area network configuration. Table 1 summarizes the foregoing discussion and provides a general overview of cost, flexibility, growth, and capacity of the various network types.

In analyzing the needs of a small business or an office automation project, these are only starting points to consider. Other important factors include the maximum number of workstations allowed, the cost of the expansion-slot adapter card required, cable characteristics and installation, maximum cable length allowed, and interfaces with existing application software. As more and more systems become available for the IBM pc, choosing the "right" system for a specific application will require increasing sophistication on the part of those involved in making the selection.

The Network Software Connection. The ultimate success of local area networks depends upon good software design. The relationship of the software required to operate LAN to its host operating system and to the application programs it supports may be the most crucial element of a local area network.

So what is network software? Generally speaking, network support software is made up of four elements: communications (networking) support, input output and file-handling support, user management support, and application program support. Figure 6 shows how network software typically fits into an existing installation.

Communications support is customarily provided by the network adapter card and is usually in the form of firmware (software that's permanently written on a ROM chip) and/or direct hardware circuitry. In the future, the trend will be to place the standard network protocols into a single chip contained on the adapter card. This is one of the benefits that will accrue from network standardization and will lead to radically lower LAN implementation costs.

One method of file-handling and input/output routine support is illustrated in figure 6. In this instance, such support is accomplished by means of a patch to the machine-independent portion of PC-DOS—the hidden file Ibmdos.com. This patch makes the network appear to the system as just another I/O device, as though it is a device supported by the machine-dependent part of PC-DOS—Ibmbio.com. As far as the Command Interpreter is concerned, any legal I/O command in PC-DOS will apply equally to remote devices served by the local network.

The next level of support is what allows users to carry out the task of managing the

network configuration. This includes such functions as attaching and unattaching shared devices, partitioning hard disk volumes, and controlling user access. This support is shown in figure 6 as a unique com file and/or a patch to the existing Command.com interpreter that works in conjunction with the patch to Ibmdos.com.

The final level of support by network software concerns the application programs to be run on the network. These might be database management systems, accounting packages, electronic mail programs, or long-distance communications programs. The main task of this type of network support is to enable application packages to make effective use of the inherent features of the local area network. The biggest challenge at this level is the management of concurrent access to shared files.

From the user's point of view, it's preferable to have little or no awareness of normal network operation. The configuration management required should be easy to use, and the need for it should be infrequent. Further, an application program should have easy access to that level of network support so that customized systems can be created. Once the hooks are made into the network, the operation of the application program should be relatively unchanged from the user's perspective. Communications protocol handling should be totally transparent to all users on the network. Such transparency is achieved if an application process uses the normal operating system commands—such as

A>DIR C:

instead of requiring something like

A>DIR C: ON SERVER1

Coming Next Month. When we resume our discussion of local area networks, we'll take a look at the practical side. In the process of acquainting ourselves with the first three LANs to be implemented on the IBM pc, we'll explore the latest in personal computer communications technology and we'll begin to develop an awareness of the applications that a network of intelligent workstations opens up. ▲

LOCAL AREA NETWORK SOFTWARE (TYPICAL)

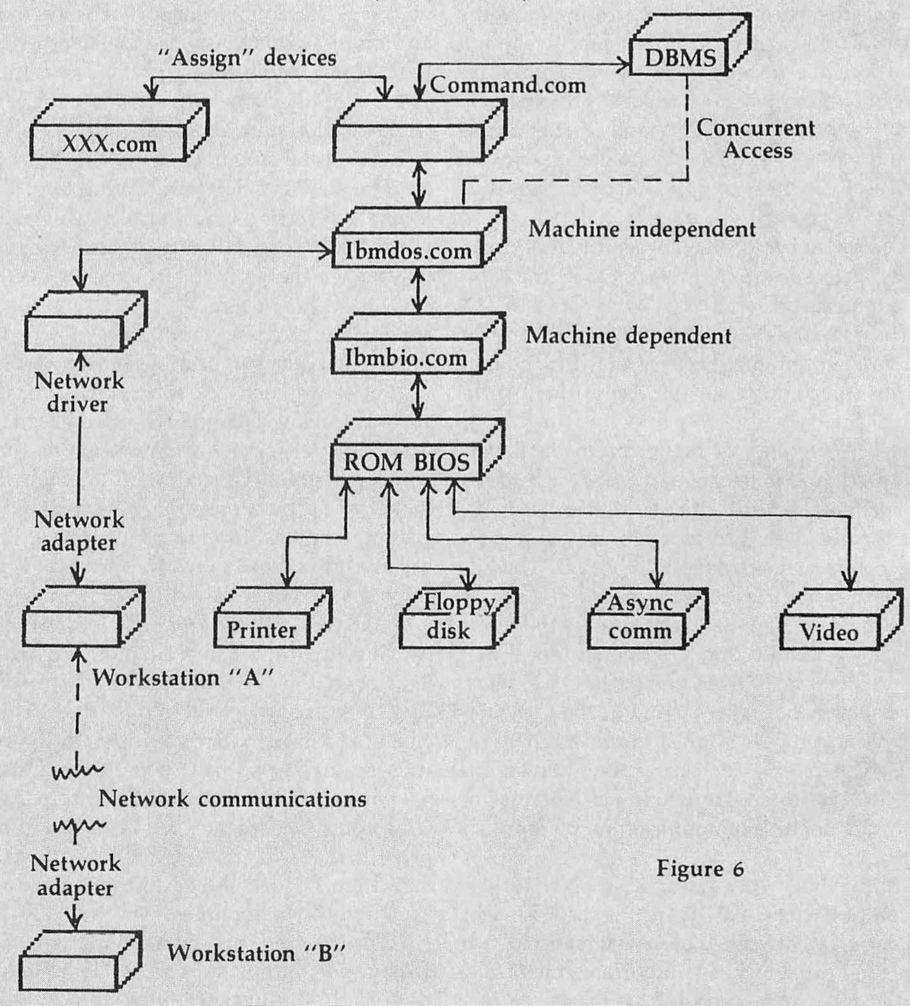


Figure 6

CP/M'83 ®
EAST

Hynes Auditorium
Thursday-Saturday
September 29-October 1, 1983
Show Hours: 10:30AM to 5:30PM daily
Sponsored by Digital Research, Inc.

THOUSANDS OF CP/M COMPATIBLES AND SOFTWARE PACKAGES ARE GATHERING IN BOSTON...

THE MOST SUCCESSFUL NEW COMPUTER SHOW IS COMING EAST.

CP/M'83 held last January in San Francisco, was the largest event organized around a single product in the history of all products. It was larger than any first or second year technology event ever held anywhere, drawing over 40,000 attendees and more than 600 displays. This September 29 through October 1, Northeast Expositions, Inc., the foremost producer of computer events, and Digital Research Inc., the inventors of CP/M, will present CP/M'83-East at Boston's Hynes Auditorium.

IF YOU DO ANYTHING IN CP/M YOU ABSOLUTELY CAN'T AFFORD TO MISS CP/M'83-EAST.

CP/M'83 is much more than a Show—it's an educational forum to help people learn about using, developing and marketing CP/M software. It's an incredibly effective medium for manufacturers to meet buyers or marketers, to meet developers. Noted leaders from the software industry will conduct the most extensive group of workshops on microcomputer software ever presented at any conference . . . ever. These informative discussions will explore CP/M applications, technical information, development aids, uses in different professions and support services. The conference program will also strive to show users how to get the most from their CP/M computers.

During each day of CP/M'83-East, from 8:30AM to 10:30AM, special industry workshops will be held for Independent Software Vendors (ISV's), distributors, dealers and manufacturers. These seminars will cover the ins and outs of developing, packaging and marketing microcomputer software. They will also offer you a chance to cultivate important industry contacts including venture capitalists.

THE WORLD OF CP/M UNDER ONE ROOF

CP/M'83-West was the largest end-user Computer Show and Sale ever; the East Coast version will be just as large an extravaganza. At CP/M'83-East you'll find everything new for your CP/M computer under one roof. In a couple of days, you can sample software, accessories and services for every conceivable application you have. The Show includes over eight thousand different kinds of products including computers, peripherals, printers, hard disks, modems, memory cards, game cartridges, video displays, and plug-in boards—plus publications, support services and an absolutely incredible array of software application packages and development aids. All the CP/M compatible hardware and software for business, industry, the professions, government, education, home and personal use, is there. And best of all, you'll save hundreds, even thousands of dollars because everything that is on display is for sale at special Show prices.

ADMISSION FEES:

One-Day Exhibits-Only ticket $10 — Tickets on sale at door — Three-Day Exhibits & Conference ticket $25

FOR MORE INFORMATION:

If you're interested in attending or exhibiting at CP/M'83-East, call 800-841-7000 (inside Massachusetts 617-739-2000). Or you can write us and we will be happy to mail you a comprehensive pamphlet which includes the schedule of seminars, workshops and conferences plus Boston hotel reservation information. Our address is 822 Boylston Street, Chestnut Hill, Massachusetts 02167.

CP/M'83-East is produced by Northeast Expositions, Inc., the foremost nationwide producer of special audience and specific product personal computer shows, including Applefest, PC'83 (for IBM PC users), The National Computer Shows, and Softcon—an International Conference and Tradefair for the Software Industry.

PREREGISTRATION FORM

☐ Enclosed is my payment for ________ (QUANTITY) Three-Day Exhibits & Conference badge(s) at $25 which admits the attendee to the Show on Thursday, Friday and Saturday and to the Conference on all three days.

☐ Enclosed is my payment for ________ (QUANTITY) One-Day Exhibits Only ticket(s) at $10 each.

Name____

Company (if any)________________________________

Address______________________________________

City__________________________ State__________ Zip__________

Telephone (area code)__________________________ ext.__________

Mail to National Computer Shows, 822 Boylston Street, Chestnut Hill, MA 02167

CP/M'83 ®
EAST

More from Microsoft

Compiler in the Key of C

by Peter Norton

When it comes to choosing a programming language, serious pc program developers have been presented with some hard choices. At the very first, IBM gave us only two languages for the pc; interpretive Basic and compiled Pascal; after a short while IBM gave us a macro assembler, and then, piece by piece, it passed out compiled Basic, Fortran, Cobol, and recently, APL. By now, almost every language known to man has become available for the pc from one source or another.

But the question of making a good choice of a programming language has not been easy. It is easy to be shortsighted, and many of us in the pc community have missed seeing the important issues related to giving our programs a solid future. The more far-sighted among us, though, have had an idea of what is needed for long-lived programs. One of the most important requirements is that programs be developed in a very "structured" way, to make them much easier to debug, change, and improve. The need for solid structured programming makes right-thinking program developers look to the three major structured programming languages: C, PL/1, and Pascal.

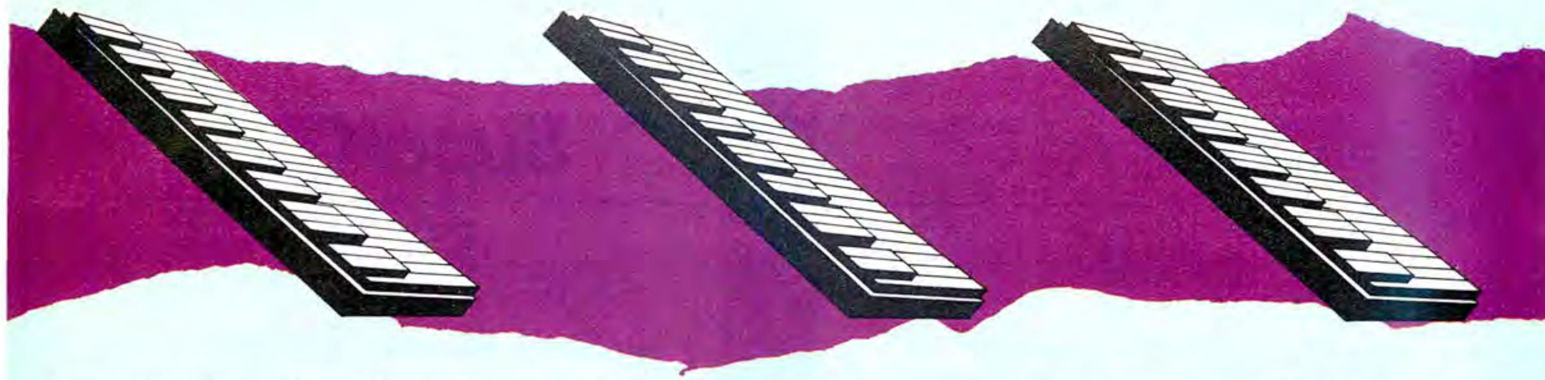
Programs with a Future. Another, even more important requirement for producing programs with a future, is that they fit into the future of our computers. This means that our programs shouldn't be exact quirks of one particular model of computer, or version of an operating system. In terms of programming languages, this means we shouldn't choose a language (and a compiler for that language), that runs a high risk of being left behind when new computers or new versions of our operating systems appear.

Since our personal computer is from IBM, and the main operating system for it—and for the host of related computers that are appearing—is from Microsoft, the sensible choice has been to stick to Microsoft's languages, particularly the ones that look like they'll be available on all IBM computers and most or all of the numerous MS-DOS computers. This is particularly important if you want to be reasonably sure that your programs will make a comfortable transition to new releases of DOS; if you choose a compiler from some hole-in-the-wall outfit, you don't really know where you might stand when each new pc and each new DOS appears. This guideline has led responsible program developers toward using Basic, Pascal, assembler, Cobol, and Fortran.

So far the common ground between these two guidelines—structured programming and Microsoft support—gave us only one language: Pascal. Pascal is a wonderful language, and it has many virtues, particularly its built-in safety features, such as its "strong typing." But no one language is best for all purposes, and each language has its drawbacks; for example, one of the drawbacks of IBM/Microsoft Pascal is that there is a great deal of overhead in the size of Pascal programs, about 18K of overhead, so that we can't write Pascal programs that aren't bulky. It's always better to have a choice, so we can pick and choose among the advantages and disadvantages of different languages. Now, thankfully, we have a choice in languages that are both structured and have Microsoft support, for Microsoft has released a C compiler.

To let you know what is most important about the new Microsoft C compiler, we'll look at it from several perspectives. First we'll see what it looks like by itself. Then we'll see how it fits into the world of C programming. Next, we'll draw a little comparison with IBM Pascal. Finally, we'll take a peek at some interesting things that come with this C compiler.

The Microsoft C compiler has interesting historical roots. Although Microsoft itself works with C, this compiler is not a direct Microsoft product. Instead, it is an adaptation of the famous and highly regarded Lattice C compiler. Lattice C was originally available for the old generation of eight-bit personal computers, and it was adapted to the sixteen-bit environment of our IBM pcs under



the wing of Lifeboat. The Lifeboat/Lattice C compiler has been available for use on IBM pcs for some time, but now that Microsoft has adopted it and "Microsofted" it, this C compiler is fully integrated into the world of DOS, where it now has a secure future in the growing world of DOS computers.

This C compiler is not an IBM product. It is strictly a Microsoft product, which means that it does not have any IBM-specific features built into it (generally none of the IBM languages do either, except for Basic). This compiler and the programs that it produces will run just fine on our IBM computers, but they will also run on any other DOS computer, which opens up a big world to any programs you develop in Microsoft C. The list price of the compiler is $500, which isn't cheap but is in line with the prices for comparable IBM and Microsoft languages.

The compiler works in two separate parts. The first part of the compiler analyzes the programs and produces any error messages. The error messages are tolerably good, but not in the best form; they don't display the part of the program that the compiler is complaining about, and they indicate only which line is in error, not where on the line. The second part of the compiler processes the semidigested results of the first part, and produces a standard "Obj" object file, ready for use with the Link program. Like other languages, C comes with a standard subroutine library; to complete a program so that it's ready to use, the compiler's object files are linked together with library routines.

This C compiler does not use a run-time module of the sort used by compiled Basic and Cobol. Compiled and linked programs are complete in themselves, and no license fee is required to distribute or sell programs. (Languages that do use a run-time module do usually require a license fee.) Parts of C programs can be separately compiled, and it is easy to incorporate assembler routines into C programs. Like most other programming languages for the pc, this compiler imposes a limit on both the program size and the data space: Each must be no larger than 64K. (Pascal, by the way, does not put a restriction on program size, but it does on the size of your data.)

The Microsoft C compiler supports the full standard C language, with only minor exceptions and changes. It also provides all the standard C subroutines and a reasonable complement of additional routines to help in the DOS environment. One chapter of the compiler manual specifically covers the requirements for converting from other microcomputer C compilers, such as Whitesmith's and BDS. Anyone who is already experienced in the world of C should have a minimum of difficulty with this compiler.

Many of the potential users of Microsoft C will be comparing it to IBM Pascal. In general, the Pascal and C languages are comparable, with a sparse, clean syntax, powerful expressiveness, and good structured logic facilities. In many ways the languages are close to each other. When their differences emerge, Pascal has a cleaner, more readable form and provides excellent safety features that aid the development of reliable programs. C is more powerful and has a more rough-and-ready format. That is the difference between the languages themselves.

The compilers accentuate the differences so that the IBM/Microsoft Pascal compiler is a smooth, polished item with excellent error messages and good programmer support, while the Microsoft C compiler is a rough workman's tool. Both compilers produce tight, fast-running code, but the Pascal compiler adds on a lot of support-routine overhead, which affects the size—but not the speed—of its programs.

In a line-for-line test program, the C compiler produced a program that was slightly tighter and ran 14 percent faster—not a significant amount but an indication of the greater ruthlessness of C. While C programs are slightly faster, the C compiler itself is *much* faster than the Pascal compiler. If storage limits are important to you, Pascal has two advantages—no limit to program size, and easy, built-in addressing of any part of memory. With C, there is a firm limit of 64K in the program size, and because there is no way to load the segment registers, assembler magic is needed to waltz around in memory (necessary, for example, if you want to work directly with the screen displays).

Included with the C compiler is the source code for two routines, one in assembler and one in C, that are a fundamental part of the start-up routines for C programs. All programming languages involve this start-up overhead, but what is unusual about this C compiler is that we are given the means to study and change the start-up routines. This gives advanced programmers the ability to control more of their programs' environment, and it gives everyone an opportunity to learn about what start-up code does.

This C compiler comes with four extra parts, which are interesting in themselves. One is the Link program, which is of little interest to us since it comes with our DOS. Another is a special disassembler program used to translate the compiler's object files into the equivalent assembler code; this is sometimes important when you need to understand exactly how a C program goes about its business. This object code listing is usually built into the features of compilers (as it is in IBM Pascal), but for C, the object listing is produced by the separate program. The third extra part that comes with the compiler is a "function extractor." The function extractor allows us to write master C programs that contain all our subroutines and then spin them off, as needed, into individually compiled modules. This function extractor seems silly, but someone must have found it important enough to have written it.

The last of the four extra parts is the most interesting. Each compiled programming language, including C, comes with a subroutine library, which is used when programs are linked. The general version of DOS includes a program, named Lib, to manipulate libraries—adding, removing, or replacing routines—but for unknown reasons, IBM has never included it in PC-DOS, so we never had the use of it. Lib *is* included in this compiler package, so anyone who gets the C compiler can use Lib, on C's library, or any other program library.

In summary, the Microsoft C compiler adds the C language to the list of languages we can count on for long-term pc program development. This C compiler produces good tight-running programs and provides a sound practical alternative to Pascal. ▲

COMSHARE TARGET SOFTWARE, INC.

ORDER FORM

	PRICE	QUANTITY	TOTAL COST
TARGET Financial Modeling	$325		
		TOTAL	
		GEORGIA RESIDENTS ADD 3% STATE TAX PLUS APPROPRIATE COUNTY TAX	
		SHIPPING	5.00
		TOTAL	$

Money-Back Guarantee
You must be completely satisfied with TARGET Financial Modeling or return it within 30 days for a full refund. This guarantee applies only to single unit end user purchases made from Comshare or from a participating dealer before Dec. 31, 1983.

☐ American Express ☐ C.O.D.

☐ VISA ☐ MasterCard ☐ check enclosed

Card number ____________________

Expiration date ____________________

Signature ____________________

Machine Type ____________________

Operating System ____________________

Company ____________________

Name ____________________

Address ____________________

____________________ Zip __________

Title ____________________ Phone __________

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

Business Reply Mail

FIRST CLASS PERMIT NO. 12239 ATLANTA, GA

POSTAGE WILL BE PAID BY ADDRESSEE

Comshare Target Software, Inc.
1935 Cliff Valley Way, Suite 200
Dept. SI-883-2
Atlanta, Georgia 30329

Multiplan™

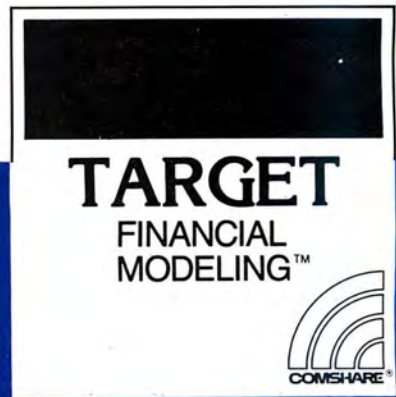
$275*

- Interactive electronic worksheet
- Maximum spreadsheet limits: 63 × 255**
- Limited English naming of cells
- Complete on-line "HELP" or reference system
- Variable individual column widths
- Extensive screen formatting (including decimals, etc.)
- Automatic forward referencing
- Limited worksheet consolidation

TARGET Financial Modeling™

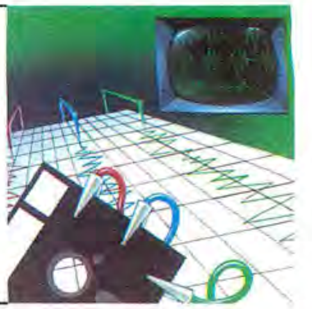
$325*

- Interactive electronic worksheet
- Maximum spreadsheet limits: 1,000 × 5,000**
- Complete English syntax for all program operations
- Complete on-line "HELP" or reference system
- Variable individual column widths
- Extensive screen formatting (including decimals, etc.)
- Automatic forward referencing
- Multiple model consolidation
- Automatic formula replication
- User defined digit significance
- Interactive display: spreadsheet or logic file
- Powerful report writer
- Allows users to build models which prompt for data during calculation.
- Money-back guarantee!



COMSHARE
1935 Cliff Valley Way
Atlanta, GA 30329
(404) 633-USER

marketalk reviews



Unless otherwise indicated, software listed runs in DOS 1.1 or 2.0 with either display adapter and requires 64K and at least one disk drive.

PC-Talk III & Smartcom II

There is little point in beating around the bush. *Smartcom II* and *PC-Talk III* are outstanding and elegant communications programs. Both programs operate flawlessly in a "dumb terminal" emulation mode. Communications parameters (start and stop bits, parity, dataword size) are easily established. "Standard" microcomputer baud rates (110, 300, 1200) are supported. (*PC-Talk III* also supports 75, 150, 600, 1800, 2400, 4800, and 9600 baud.) Tests of standard terminal operations at 300 and 1200 baud under PC-DOS 1.10 and 2.00 produced no problems or complications for either program.

Both programs have resolved a serious problem that plagued the first release of IBM's *Asynchronous Communications Support.* Many mainframe computer systems require the presence of a destructive backspace to correct errors. In *PC-Talk III*, this operation is accomplished with a control-backspace. In *Smartcom II*, the delete key has the same effect.

ASCII file uploads and downloads were tested between the pc and the CDC Cyber 730 (operating under NOS) and between the pc and a DEC PDP 11/70 (operating under RSTS/E) at 300 baud. Additional upload and download tests for ASCII files were performed between the pc and the Source at 1200 baud. No problems of any kind were encountered with either program.

The operations just described are typical of most good microcomputer communications programs. It is in the enhancements to these basic operations that *Smartcom II* and *PC-Talk III* really shine. It is clear that the programs were designed by people with a genuine understanding of the needs and problems of microcomputer users.

A frequent problem, for example, is receiving information that you decide you want to store on disk before you have opened a log file to store it. With most communications programs, such information is simply lost, and it must be retransmitted to be saved in a file. *PC-Talk III* contains an ingenious alternative. Any time you type alt-S, the current screen display is immediately copied to a file called Scrndump.pct. Subsequent invocations of this feature append the new information to the Scrndump.pct file. Thus, it is possible to function without opening a log file at all. You may save any information you wish *after* you have seen it and decided that it is worth saving.

Another frequent problem is the need to review dialog that has transpired earlier in the session. *Smartcom II* routes all input and output to a special buffer. It is possible to read through up to about 4K of your prior dialog by using the PageUp and PageDown keys to move a screen at a time, or the up and down arrow keys to move a line at a time.

Both programs allow the user to define macros (sequences of keystrokes that are invoked by a single keystroke). In *PC-Talk III*, up to forty macros of up to 126 characters each may be generated and saved on disk. These macros are referenced to the function keys alone or in combination with alt, control, or the shift key. Up to ten additional temporary macros of up to fifty characters each may be referenced to alt-1 through alt-0. The temporary macros are not saved from session to session.

This macro capacity is useful for a number of purposes. For example, if one is performing a literature search through a service like Dialog or Knowledge Index, the search commands may be assigned as macro definitions. One may then invoke a complex search strategy any number of times by using single keystrokes. The macros may also be used to store passwords and log-on sequences, making logging on to a mainframe a semiautomatic process (dial the phone number, then strike the appropriate macro key). Frequently used commands on services like the Source (*post scan IBM*, for example) may be stored and easily invoked.

Smartcom II takes the macro concept a step beyond *PC-Talk III*. This program allows the assignment of fully interactive sequences of macros. Up to sixteen lines of up to forty-eight characters each may be assigned. Each line is sent when *Smartcom II* sees a user-definable prompt from the remote computer, or when a user specified time-out is exhausted. The macro set is invoked by typing an alphabetic key with alt (or by using the sequence F5 followed by an alphabetic key). Up to twenty-six such macro sets may be assigned to each of up to twenty-six different communication sets.

Among the many uses of the *Smartcom II* macro capacity is the ability to have completely automatic log-on procedures for communication with mainframes. Since each communication set may be associated with an automatically dialed telephone number, it is only necessary to invoke the communication set (with two to four keystrokes). The number is automatically dialed, and the log-on macro set is automatically invoked with no further user intervention.

As might be expected, both programs take full advantage of the Hayes 300 and 1200 baud Smartmodems. *PC-Talk III* can store up to sixty telephone numbers for automatic dialing. *Smartcom II* can store up to twenty-five numbers. In both cases numbers are stored with descriptive titles. Both programs have automatic redialing routines that make repeated attempts to establish communication in the event of a busy signal.

With *Smartcom II*, each phone number is stored in a communication set that also includes up to twenty-six macro sets. With *PC-Talk III*, phone numbers are stored independently of the macros and any stored macro may be used with any phone number.

Communication at 1200 baud and above with standard telephone lines is much less reliable than communication at 300 baud. The major culprit seems to be noise on the phone line that the computer interprets as data. This noise can generate garbage in the received file, or, perhaps even more disastrously, it can produce characters that are acted upon by the computer in unanticipated ways. An inappropriate control-Z in the middle of a file, for example, would signal to DOS that the end of the file had been reached. It would then become difficult if not impossible to read any information beyond this point.

Neither program can protect the user against the first problem. If "Hi, sailor" is received as "]Hi, sailor" or "Hiz, sailor," the inappropriate characters cannot be filtered out by any known com-

munication program. On the other hand, there is some defense against unwanted control characters or characters outside of the expected ASCII range of data. *Smartcom II* contains a "formatted" file transmission mode that strips out troublesome control characters. *PC-Talk III* allows the user to select up to three ASCII characters to be filtered out or converted to other characters. In this way, the most damaging characters can be filtered out. This facility can also be used to strip out any other unwanted character that the remote system sends as a matter of course.

The unreliability of 1200-baud transmission might lead some users to fall back to 300 baud for transmission of critical data. However, this drastic step may not be necessary. Both programs contain special verification protocols that allow a substantial increase in the accuracy of transmission of information. *PC-Talk III* allows the use of Ward Christensen's XModem error-checking protocol. *Smartcom II* uses the "Hayes Verification Protocol."

After repeated unsuccessful attempts to transmit a long ASCII file in the normal mode, the verification protocols were invoked for each program. Attempts to use these protocols at 1200 baud were completely successful. Error-free transmission was achieved on the first try with both programs.

These verification protocols should be a cause for celebration. Unfortunately, the programs use different and incompatible protocols. Since both the sending and receiving verification protocols must be identical, this means that *PC-Talk III* and *Smartcom II* cannot communicate with one another in verification mode. These programs are so good that they will probably become the most widely used communications programs in the pc community. It is a shame that they are incompatible at this level. One can only hope that future revisions of either or both programs will resolve the problem.

Both programs have a full complement of file operations, including printing, deleting, uploading, downloading, change of default drive, file directories, and file display. All these operations may take place while you're on line, without severing the communication line. In addition, both programs allow pacing or line-by-line uploads. This capacity is important for communication with some computer systems that have buffers too small to receive the transmitted file successfully. Slowing down transmission rate allows such computers to keep up with the information flow.

An important feature of both programs is the ability to transmit binary files (com files and exe files, for example) in addition to the usual ASCII transmission capability. An implication of this feature is that one may send an exe file directly instead of transmitting a hex file in ASCII that must then be converted to binary with a special program.

Smartcom II can function in what Hayes calls remote operation for file uploads and downloads. It is possible to leave your computer set up with the program and call in with another computer that is also operating with *Smartcom II*. Your computer will answer the phone and allow you to transfer files in either direction via commands from the remote computer. Password protection is available in this operating mode.

Both programs are easily configured for various hardware combinations. Com1 and Com2 are supported for communication or printing. Serial or parallel printers may be used.

Documentation is complete, clear, and informative. Hayes, in particular, deserves special recognition for the quality of the *Smartcom II* manual. It is not only a model for program documentation, it is also a miniature introductory course on communications.

The most obvious difference between these programs is price. *Smartcom II* sells for about $120 and is easily worth the money. *PC-Talk III* is marketed through Andrew Fluegelman's Freeware system for a suggested contribution of $35. The dollar value of this product is startling. It easily outperforms programs selling for more than five times its price.

It is impossible to recommend one of these programs over the other. Users who can afford both programs should have them in their software libraries to take advantage of both verification protocols and to enjoy the unique features of each program. Users who can afford only one of the programs will be pleased regardless of their choice. These programs represent the state of the art in communications for the pc. BA

PC-Talk III, by Andrew Fluegelman, Freeware (Box 862, Tiburon, CA 94920). $35 (suggested contribution).
Smartcom II, by Hayes Microcomputer Products (5923 Peachtree Boulevard, Norcross, GA 30092; 404-449-8791). $119.

California 10 Pak

Ask most people what they think of when they hear "utility" and they will reply "power." That's what CSPI brings you with its California 10 Pak of utilities: power for the IBM pc. The double-handful of programs includes some that are merely useful and a trio that qualify as indispensable.

Disasm will disassemble areas of memory, com or exe files. Unlike the unassemble command in Debug, *Disasm* will write 8088 mnemonics directly to disk in ASM format. An important addition

is the use of labels to reference calls, jumps, and moves so that additions, deletions, and changes may be made to the file. Certain ambiguous statements, such as *return far* or *return near*, are listed as defined bytes with clarifying remarks. The ASM file can be reassembled with either the macro or small assembler.

The program is indispensable for finding how a specific program is put together or for learning 8088 assembly language. What's more, the use of labels means that the program won't be destroyed by changes made in the source code.

There are two items lacking in *Disasm* that are found in similar Z80 disassemblers. One is a remark column showing the ASCII equivalent of program code so that data may be more easily recognized as distinct from program code. The other is an adequate statement for labels that reference addresses outside of the program module or area of memory being disassembled. Despite these two omissions, *Disasm* is an invaluable tool for any pc assembly language programmer.

Ccomp compares two files to show their differences. It's a vast improvement over the DOS *comp* command in that it shows all of the differences between the files—not just the first ten. Nor need the file sizes be identical for *Ccomp* to work.

All changes are shown in context, each file in a separate screen window with differences displayed in reverse video. Data can be echoed to the printer—as it can from most of these utilities.

Cdiff is similar to *Ccomp* but displays the differences between two *text* files. An excellent feature of these programs is their automatic resynchronization. *Ccomp* can look ahead as many as one hundred lines so that insertions and deletions in the files won't throw the comparisons out of whack. A plain vanilla comparison of *dBase II* files, for example, would list everything from the point of insertion or deletion as differences; *Cdiff* overcomes this problem.

The files need not be the same length, so that material appended to a file *will* be displayed with *Cdiff*.

Cdir permits a selective display of directory entries or a full sorted directory that includes hidden files. It can be presented in ascending or descending order by filename and suffix, suffix and filename, file size, or date and time.

Cdisk marks the File Allocation Table to show free space, files, and locked out, reserved, or cross-linked clusters. The location of separate user files is shown, and *Cdisk* can be toggled to give a hexadecimal or ASCII display of any disk sector.

Cfile and *Cmem* provide combined hexadecimal/ASCII displays of any file or area of memory. The *Browse* utility shows full pages from a text file.

Csort will rearrange an input file and create an output file in the desired order. It includes a sort within a sort nested up to five levels and permits viewing the sorted data on the monitor.

Cenv reads the equipment switches and displays the peripheral devices attached to the pc. It shows the data from the disk parameter table and the addresses used by the current interrupt vector table. A map of the memory allocation is shown, with areas marked for Basic, BIOS, DOS, system files, the color and monochrome monitors, and user RAM.

Most of the programs use the PageUp and PageDown keys to move forward or backward, and Home and End to move to the extremes of the information displayed. PrtSc is used to provide hard copy, and all of the programs are compatible with DOS 1.1 and 2.0.

For all of the software written for the IBM pc, there is still a woeful shortage of utility programs. *California 10 Pak* is a welcome addition, with its abilities to find all the differences between program versions or data files. A disk-labeling disassembler will be a welcome addition to any programmers library. DR

California 10 Pak, by California Software Products (525 North Cabrillo Park Drive, Santa Ana, CA 92701; 714-973-0440). $100.

Keyfixer

This is a set of five plastic collars that are easily glued over some of the problem keys on the pc and XT keyboard. The enter key becomes an accessible one-and-a-quarter inches long. The two shift keys become a quarter inch wider. The control, backspace, or any two other notched, dark grey keys (it's your choice) grow a half inch. Debate is still raging as to whether these are physical or psychological improvements; they're probably a combination of both. If you watch your weaker fingers hit any of those critical keys, you'll know if Keyfixers are needed. Silicon adhesive included in kit. Collars can be removed with no damage to the original keys. MF

Keyfixer, by Vertex Systems (7950 West Fourth Street, Los Angeles, CA 90048; 213-938-0857). $14.95.

Microsoft Decathlon

Even people who take little regular interest in sports often feel a thrill upon hearing the opening strains of ABC's Olympics coverage theme and seeing the nations of the world, as represented by their finest amateur athletes, join in parade. If you're of this mind, *Microsoft Decathlon* is directed to you as well as to sports buffs.

The game, upon booting, comes as close to purveying the mood of Olympic pomp and ceremony as a computer program probably could. An athlete pulling a title word runs on-screen, reminiscent of the Olympic torch runner. Another athlete puts the shot to dot the "i" in Olympics. Then a whole row of runners, divers, and vaulters are animated across the top of the screen, each symbolizing one of the challenges of the decathlon.

What's in store then is one of the most unique game programs yet devised.

Decathlon presents the series of ten competitive athletic events that make up the modern version of the ancient Greek track-and-field contest. Real decathlon athletes must compete in all ten events during the span of two days. The events are the 100-meter dash, the long jump, the shot-put, the high jump, the 400-meter dash, the 110-meter hurdles, the discus throw, the pole vault, the javelin throw, and the 1,500-meter run.

This is precisely the content of the program. As many as six people can compete through the ten events with each other, each choosing his country of fictional origin. They also compete with the record-breaking score by Bruce Jenner at the 1976 Olympics.

There is a large measure of unreality inherent in trying to simulate athletic contests on the pc keyboard, but, to the extent possible, the author has done it. For example, running is simulated by two fingers on the keys, perhaps the least accurate of the simulations. All the rest concentrate on the timing involved in executing the event and on the judgment of the relative position. Whether or not these are accurate simulations of the factors involved in each sport is difficult to say. That they are part of playing the game is a fact. Each event is unique, and you'll find that each person you play with will probably excel in a different area.

Although *Decathlon* isn't addictive in the ways that *Crossfire* and backgammon are, it presents a pleasant challenge that will draw you back to the game many times. Outstanding graphics, sound, a practice mode, and animated athletes that move like real people add to the game's enjoyability. Of note in this regard are the hurdler and, in his Fosbury Flop jumping position, the high jumper.

At the end of the competition, the flags of the first-, second-, and third-place countries are displayed. Whichever flag you pick to perform under, you won't want to miss the feeling of standing in the winner's circle and hearing the cheers of the crowd, even if it's only in your mind. MCT/MF

Microsoft Decathlon, by Timothy W. Smith, IBM (Boca Raton, FL 33432; 800-447-4700).

Beneath Apple Manor: Special Edition

The original release of this graphic version of *Dungeons & Dragons* has been an Apple classic for years. Author Don Worth has since brought it up to date with graphics, speed, and such other bells and whistles as a *save game* function.

He also added a few treasures and fancied up the ending. For the pc release, he added ten more rooms in monochrome for the hard-core purist to play in. And there, in his wisdom, he stopped. The original game, with all its super features, remains the same.

Beneath Apple Manor is, in this time of gaming complexity, an exercise in simplicity. Explore the dungeons, fight the monsters, build your character, go deeper, and do it all over again.

But simplicity can be another way of saying elegance. Consider that every dungeon level is generated randomly, every one is one you've never seen before. And you won't see it all until you explore it, bit by bit. Then, of course, unless you take the memory wipe-out potion, all that you've explored will stay on the screen, so that you can see the whole thing eventually. Exploring and discovering each maze is a lot of the intrigue of *BAM*; what will the next dungeon look like?

There are tradeoffs. In graphics, you're limited to five rooms per dungeon; if you choose more, you play in mono. And one of the new treasures is a potion that reveals the whole level you're on. That's undoubtedly supposed to be a boon, but there's one reviewer that won't chance taking that one again.

The strategy challenge remains, and there's plenty of it. Few *BAM* players haven't a clear-cut strategy they'll remember above all others. Experience points are tradeable for character points in any of a relatively standard four areas—strength, intelligence, dexterity, and body (constitution). Because new levels are populated on the basis of your average and because monsters vary in how they affect you and in what affects them and in which levels they tend to cluster, where you put your points at various times during the game is of strategic importance.

Its eternal newness renders *Beneath Apple Manor* addictive. Even after you reach the bottom of the dungeon and walk away with the prized golden pc, the challenge of doing it again at a higher level of difficulty—there are ten—remains. Some fans of the early *BAM* have changed the rules to keep playing; they never take the golden computer when it's offered. Their goal is to see how much of a superhero they can build. Since the dungeons keep getting harder to match your strength, the challenge continues.

BAM doesn't have the complexity of a *Wizardry* or *Temple of Apshai*, but it remains a light, interesting, thoroughly enjoyable game with a wide enough range of play levels to be fun for very small children and to challenge veteran adult players. MCT/MF

Beneath Apple Manor: Special Edition, by Don Worth, Quality Software (6660 Reseda Boulevard, Suite 105, Reseda, CA 91335; 213-344-6599). $29.95.

muMath

muMath is a symbolic mathematics (computer algebra) package for microcomputers. It has been available for CP/M and TRS-80 computers for some time and has recently been released for the IBM. It is an extremely powerful and useful package for professionals and students alike.

Mathematical computations on computers are done primarily by what are known as "numerical methods." These are methods for approximating solutions to mathematical problems on computers. Symbolic mathematics is different. People use symbolic mathematics to solve problems. Symbolic mathematics has the advantage that it is exact and uses the conceptual framework of standard mathematics. Consider this configuration:

```
4 * X + 4 = 0
```

There are many numerical techniques for solving this simple equation. For example, we might use the Newton-Raphson iteration technique by first determining the derivative and proceeding iteratively in the direction of the solution. People, however, would go about solving this problem in a fundamentally different fashion. First they might move the 4 to the right side of the equation by subtracting 4 from both sides. Then, by dividing by 4, they would obtain the answer: −1. The latter method is an example of symbolic mathematics, since we manipulate symbols rather than numbers to arrive at the solution.

muMath uses symbolic mathematics. It is written in an artificial intelligence language called muSimp, which is an outer-syntax parser for an internal Lisp structure. It performs exact arithmetic and symbolic manipulation, just as people do (without the errors). For example, if you ask *muMath* to add 4/15 and 1/15, it will compute the answer as 1/3. Not 0.333333, but exactly 1/3. The result can be printed with as many decimal places as desired in decimal format, or it can be printed in exact rational arithmetic terms.

The arithmetic capabilities of *muMath* are quite impressive. One can work in any base (from base 2 to base 36) and work exactly with very large and very small numbers. Ever want to know what 100 factorial is? *muMath* will compute the result in a couple of seconds:

```
9332621544394415268169923
8856266700490715968264381
6214685929638952175999932
2991560894146397615651828
6253697920827223758251185
2109168640000000000000000
000000000
```

muMath can work with numbers much larger than this. Want to know what this number raised to the twelfth power looks like in binary? *muMath* can tell you, easily and quickly.

When working with fractions, *muMath* automatically performs whatever simplification it can. For example, if you ask it to add 153/2142 and 78/546, it automatically simplifies the answer to 3/14.

This is just a small sample of the power of the program. *muMath* can perform algebra, such as solving equations, simplifying or expanding expressions, and matrix algebra. The matrix algebra manipulations available include addition, subtraction, division, transposition, expansion of determinants, construction of an identity matrix of any size, and the finding of the inverse of a matrix. The program even supports ragged arrays.

It will also do symbolic differentiation and integration. Partial derivatives of exceedingly complex expressions can be computed. It will properly differentiate virtually any expression you can come up with. *muMath* also comes with packages to determine the limit, Taylor series, repeated sum, or product of an expression. These packages are not only useful for your homework and theoretical studies, but also for the evaluation of otherwise intractable integrals. Often these problems can be put into a tractable form by applying one of these packages.

So far we've only considered *muMath* in its "calculator mode"—and what a calculator it is. It also has a programming mode. In this mode you can write programs in *muMath* to perform mathematical tasks, or even artificial intelligence tasks.

The language in which *muMath* was written is called muSimp. It comes with the *muMath* package, so you get both for the price of one package. muSimp, as mentioned previously, is really a cortex parser for an internal Lisp interpreter. One of the things most people find difficult about Lisp is Lots of Irritating Silly Parentheses. muSimp provides an external structured syntax more like Pascal than Lisp. Programs in muSimp consist of functions and subroutines built up in a hierarchical fashion. Finally, one function is written that makes appropriate calls to all the necessary smaller functions.

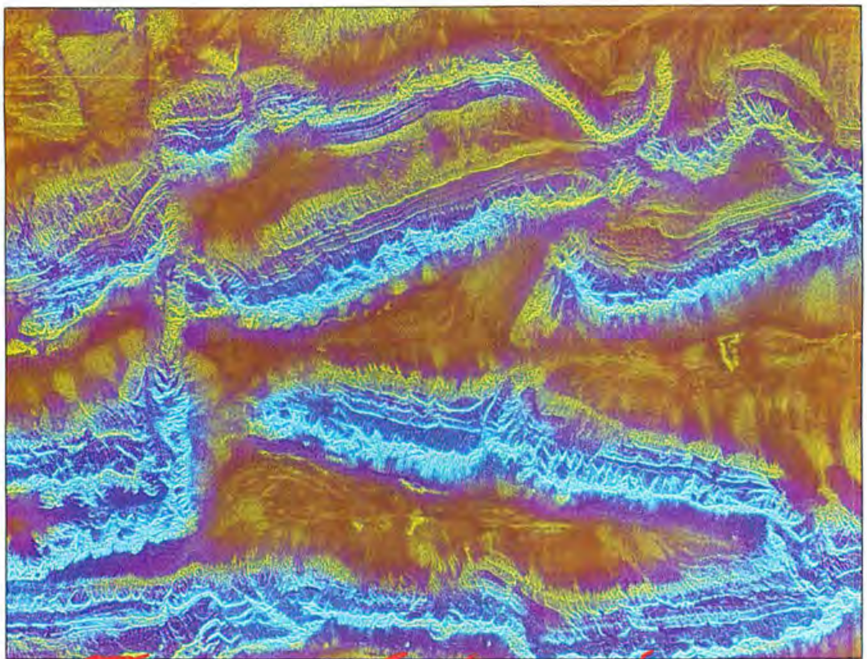
muSimp also has a trace package and a resident function editor. There wasn't room for these features on the CP/M and TRS-80 versions of muSimp, and they are a welcome addition. The editor is simple to use and quite powerful.

The manual is well written. It describes the packages and their use quite well. It has a complete table of contents and two indexes. The weak point of the manual, if there is any, is in the explanation of the programming language. Although muSimp is similar to RLisp, there is no book currently in existence that is a tutorial on muSimp or RLisp. A large section of the manual is devoted to a description of the muSimp language. A more in-depth tutorial would be nice. On the other hand, with the *muMath* package, you do receive a disk of tutorial lessons for both the calculator and programming modes for *muMath*.

With the package, you get the manual and two disks of *muMath* packages. The language interpreter is a com file, and the rest of *muMath* is stored in ASCII files. You can pick and choose which parts you want to have resident in memory at once, saving space. *muMath* is quite simply the most powerful and versatile package of its kind for microcomputers. It can be used by students, teachers, scientists, mathematicians, engineers, or anyone else working with mathematics. Computer algebra will not replace numerical mathematics. There are just too many problems that don't have closed-form solutions or that are better handled by numerical methods. *muMath*, however, provides unprecedented power for microcomputer users for a very reasonable price. BPD

muMath, by The Soft Warehouse (Box 11174, Honolulu, HI 96828; 808-734-5801. $250.

THE PC SUPPORTS THE SPACE SHUTTLE



Creating an IMAGE

by Craig Stinson

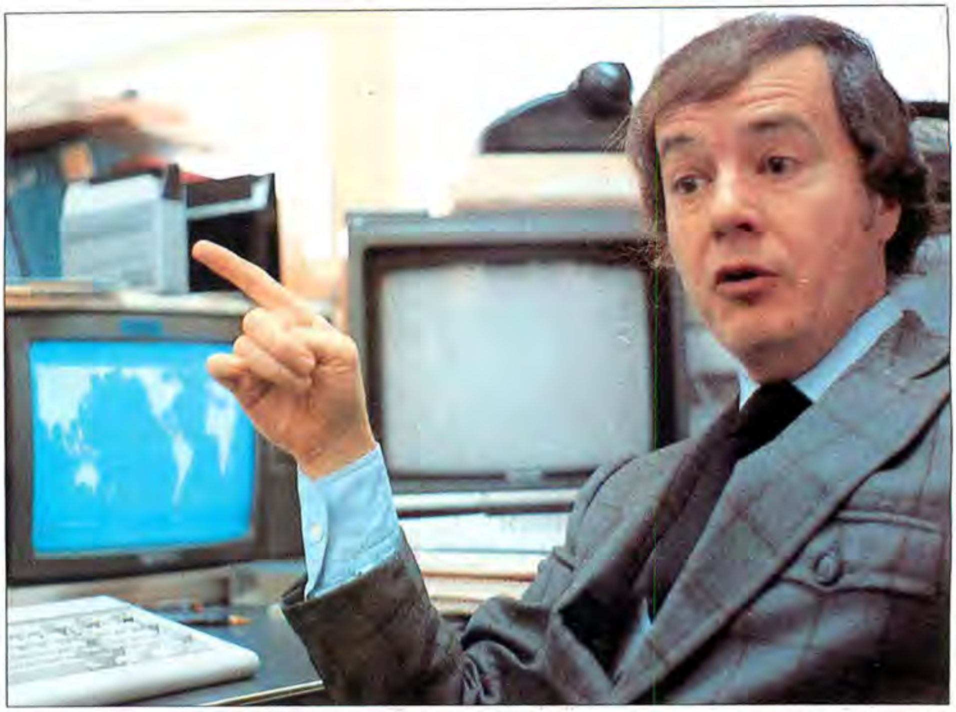
If you've visited Los Angeles recently, you've undoubtedly noticed that the airport is in a state of, shall we say, transition. New terminals and parking structures are going in, old roads are coming out, tempers are short, delays are long, patience is thin, and wits seem intractably thick.

It's not just the airport. The whole city is enthralled in a crescendo of noisy preparation for the event that will overwhelm it next summer: the 1984 Olympic games.

Nestled in a hillside above the northern perimeter of the city, a group of scientists

(Opposite page) This radar image of a swampy coastal region of New Guinea was taken by SIR-A in 1981. False colors were applied in the laboratory to the original black-and-white image. Oceans and rivers appear as dark blue. (Above) This false-color radar image of an arid, mountainous region of the Sinkian Province in northwestern China was taken by SIR-A in 1981.

Research for this article was contributed by Marsha Stewart and Paul Mithra.



Harris's ground-track program uses the pc's color graphics to superimpose the path of the shuttle over a map of the world.

and engineers at the Jet Propulsion Laboratory are preparing for a different kind of event, one that will take place at about the same time and that will be of equal, if not greater, moment: the seventeenth flight of the space shuttle. In contrast to the hoopla below, the work here is quiet and serious. A lot will ride on this mission.

To be specific, what will ride on it are two experiments designed to enhance our knowledge of the earth's topology and composition. One is a photographic mapping experiment that will employ a large-format camera. The other is the second outing of the National Aeronautics and Space Administration's Space-Imaging Radar (SIR) project. And in support of the latter experiment, your favorite microcomputer is playing an uncommonly significant role. A series of programs written for an 8087-equipped IBM Personal Computer will be used to monitor and command the radar equipment aboard the shuttle.

The radar-imaging equipment in the SIR project will direct radio frequency energy from the space shuttle at selected earth targets. Scientists will record the reflections of those radio beams from the targets back to the spacecraft. The reflection patterns will be subjected to various kinds of analysis and ultimately converted into images like the ones in the pictures accompanying this article: the photographs (radiographs?) shown in these pages were produced by the earlier episode in this SIR project, which rode the second flight of the shuttle in 1981.

Radar imaging has some advantages over ordinary light imaging (photography). The first is that it doesn't require any ordinary light. The experiments will, in effect, generate their own illumination from the spacecraft and will be able to image any portion of the earth passing under the shuttle, even if it's facing away from the sun. Second, clouds are transparent to radar; certain kinds of ground features are also relatively transparent. The first SIR mission, for example, was able to yield valuable information about the topology of the earth *beneath* the surface of the Sahara Desert, because the sand there was dry enough to allow penetration of the radar energy. Finally, the experimenters say that radar images are qualitatively different from light images, that scientists can often get information from them that is complementary to what they can obtain from ordinary photography.

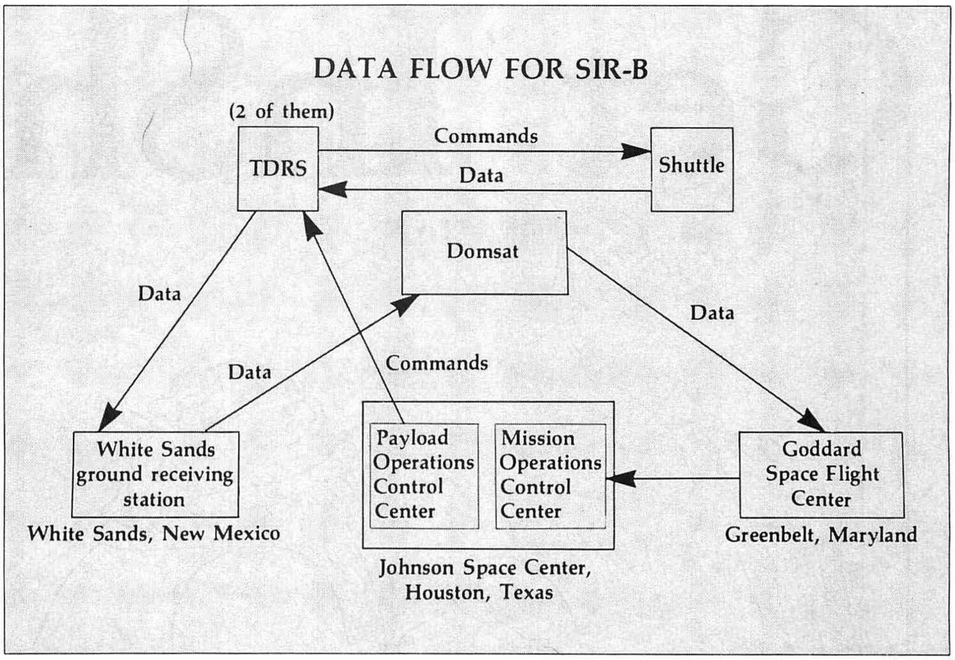
The SIR experiment that will fly next year (called SIR-B to distinguish it from the first outing, SIR-A) is much more ambitious than its predecessor.

"In 1981 the technique was new, and we were very interested just to see that everything worked as we expected it to. We turned the instrument on over land and we were happy with the results we got," says Henry Harris, mission-design manager of the project. "Anything we got was useful. This time we're going more toward specific sites, and trying the more difficult maneuver of imaging the same site from different angles on successive passes around the earth."

A crucial new feature of the current

phase of the experiment is the use of what's known as a synthetic aperture. This is a way of achieving higher-resolution pictures.

The maximum resolution of radar images is dependent on, among other factors, the size of the antenna that picks up the reflected radio pulses. As a way of increasing antenna size far beyond what would be possible with a simple physical device, the SIR-B experiment will take advantage of the shuttle's motion through space; the path of the instrument along a certain stretch of its orbit will be treated as though it were a gigantic antenna. This will make it possible for the experimenters to "see" more more detail on the earth's surface, but it will make the process of converting incoming data into visible images that much more complex; scientists will have to take into consideration factors of both time delay and Doppler shift, since the signals returning from a given point on the earth will, in effect, be chasing a receding spaceship.



The data will be analyzed at the Johnson Space Center in Houston, as well as at JPL. To get to Houston, it will take the long way, passing first to either of two tracking and data relay (TDRS, pronounced teedrus) satellites that will be positioned in geosynchronous orbits, thence to a big-dish antenna at White Sands, New Mexico, from there via communications satellite (Domsat) to the Goddard Space Flight Center in Maryland, and from Maryland over a "hard link" to Houston (see diagram). The journey, of course, will take less time than it took you to read the preceding sentence.

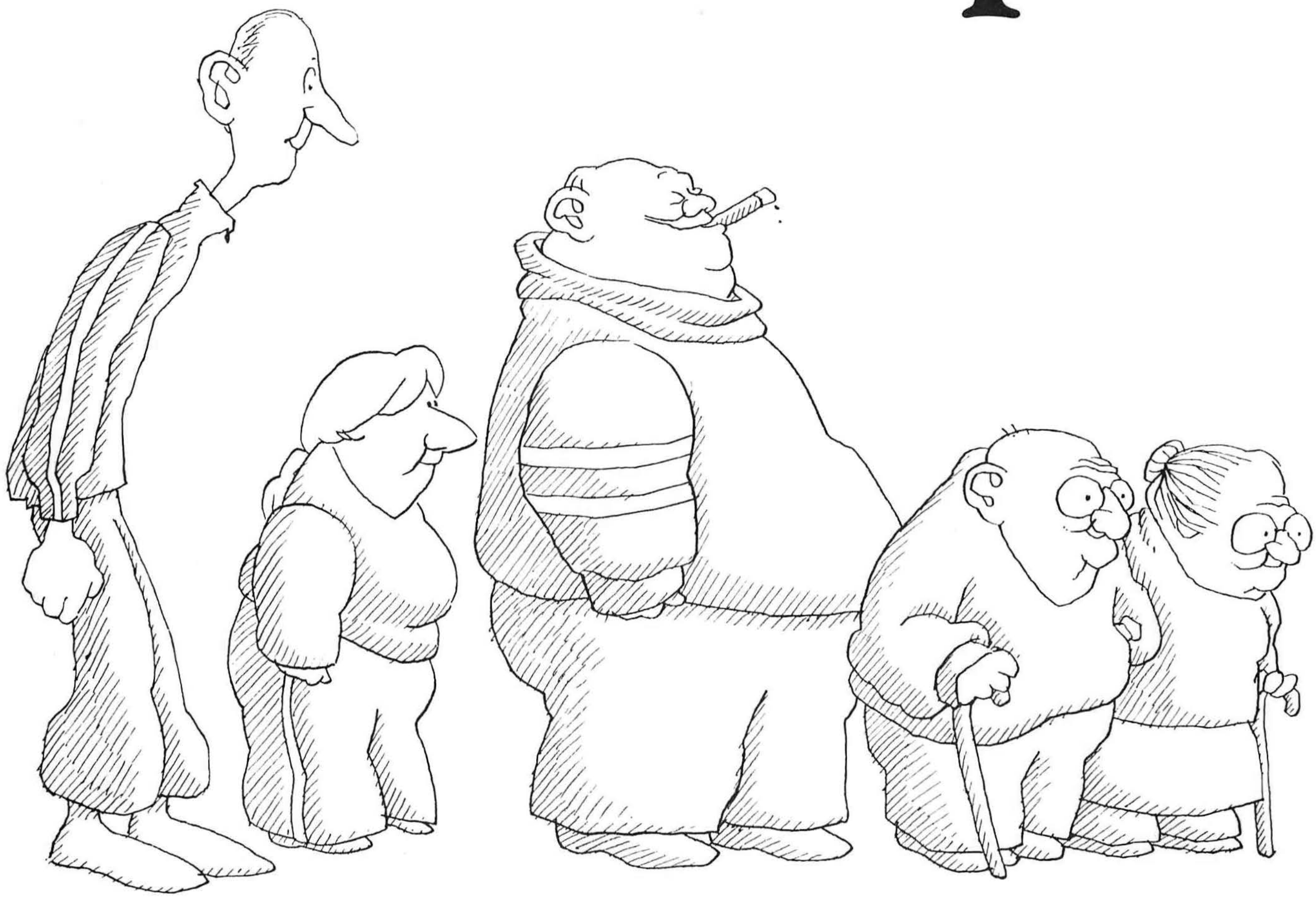
Turning raw data into pictures, however, will take a little longer.

During each second of its operation, SIR-B will image an area of the earth about seven kilometers by one hundred kilometers, at a resolution of about twenty-five meters. Approximately a million input pixels will be arriving each second, each of which will have to undergo about a thousand computer operations en route to becoming an output pixel. To process this volume of data in real time would thus require about a billion operations per second. JPL hopes to have a processor of such capacity and speed by 1985; in the meantime, plans are to process fully only about two hours' worth of SIR-B's data during the year following the week-long flight; the rest will be worked up over the course of the two years following.

In case you're wondering, the IBM pc will not be involved in this massive post-mission numbercrunch.

But it will be busy indeed during the flight. A linked pair of pcs will be stationed in the Payload Operations Control Center at Houston, doing four essential tasks: calculating the ground-track of the shuttle (its position relative to the earth at any point in time); verifying position by "taking a slice" of the incoming signal from time to time and subjecting it to rudimentary waveform analysis (such analysis can tell the experimenters whether the signals they're receiving are reflected from water or from land); monitoring the health of the equipment—such things as its power, temperature, antenna angle, pulse repetition frequency, and gain; and developing commands to reconfigure the instrument for particular targets along the way or to correct error conditions, should they arise. The astronauts aboard the shuttle will be able to attend to the radar equipment if for any reason the crew in Houston should lose contact. But, for the

Get InShape!™



The personal diet and exercise program for IBM-PC users.

No matter what your fitness formula is, InShape™ and your IBM-PC can help take the work out of tracking your diet and exercise habits. Whether your racket is tennis or you're keeping tabs on the dinner table, InShape helps you measure your progress on the road to personal fitness.

The Waist Watcher

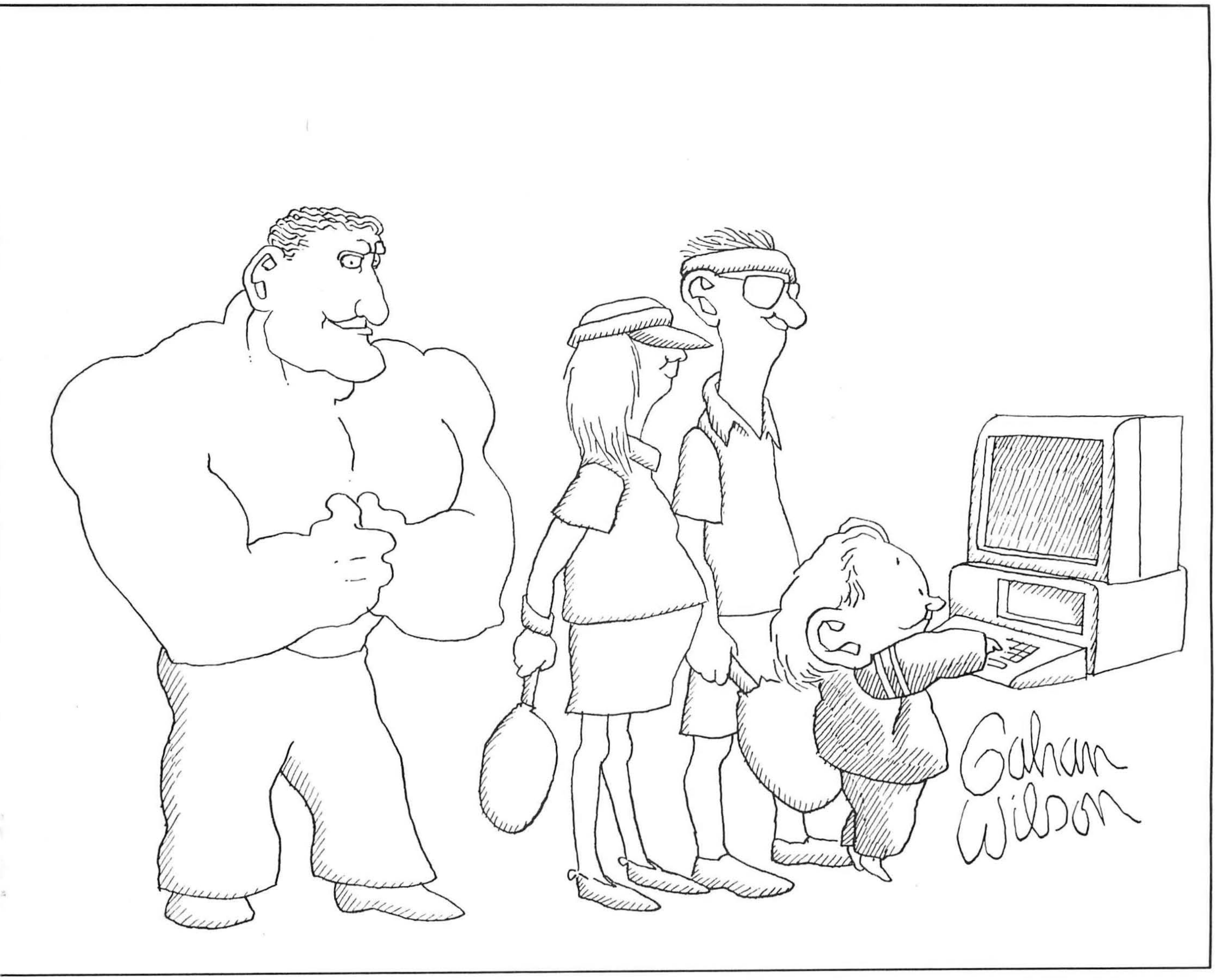
Keeping accurate daily records of what you eat is important, regardless of whether you're dieting or simply concerned about proper nutrition. InShape makes it easy. InShape's computerized food dictionary lists approximately 1,000 different foods, along with their food group and calorie, carbohydrate, protein, and fat content. The food dictionary can easily be edited and new foods added.

To update your daily record, simply select the food from the dictionary, then specify the number of servings and the meal — breakfast, lunch, supper, or snack. Over 50 entries per day can be recorded. InShape automatically totals your daily consumption by meal and food group.

And when it comes to exercise...

Of course, InShape can't do your exercising for you. But it *can* make getting started and staying with it a lot more fun.

Using the standard aerobics formulas, InShape automatically calculates points for 24 aerobic activities — from running to racquetball. Merely enter information pertinent to each exercise performed. InShape figures the points you've earned and keeps track of your weekly total.



Spotting trends

With InShape, you can *see* exercise and eating patterns over a period of time. Flexible plotting capabilities let you display up to a year's worth of information and view it on a daily or weekly basis. InShape can show you such detailed information as calories consumed compared to aerobic points, percentage of protein eaten at each meal, carbohydrates consumed by food group, and much more.

The main ingredient — you

Once you've set your diet and fitness goals, InShape helps you pinpoint problem areas. And helps measure your progress as you build a computerized personal fitness and nutrition profile. All you need is willpower. And the desire to get in shape!

For the introductory price of $95, *you* can get InShape. For more information write DEG Software; 11999 Katy Freeway, Suite 150; Houston, Texas 77079; 713/531-6100. To order, call 800/231-0627 (continental United States, excluding Texas).

DEG software
11999 Katy Freeway, Suite 150
Houston, Texas 77079
713/531-6100, 800/231-0627

Available at participating retail stores.
Dealer inquiries welcome.

Requires IBM-PC, minimum one disk drive, and 80-column monitor. Printer optional. Compatible with DOS 1.0 and 1.1 with 64K memory, or DOS 2.0 with 96K memory and double-sided disk drive.

most part, the humans aloft will be otherwise engaged; if all goes well, SIR-B will be controlled entirely from Houston.

The software to do all this was written by mission-design manager Harris, with the help of one other programmer. Harris, who admits to such off-duty activities as writing adventure games and playing piano in Los Angeles-area jazz clubs, designed the SIR-B mission programs entirely in PC/Forth, supplied by Laboratory Microsystems. His use of Forth brought down no small amount of flack from certain quarters at JPL. Harris defended his language choice in a detailed memorandum to his critics, citing the language's extensibility, structure, and readability, and pointing out the slowness, the reputed unreliability, and the lack of 8087 support of IBM's Fortran. Readability is not generally put forward as a selling point for Forth (because the language uses reverse Polish notation and allows programmers to define and use command verbs on the fly), but Harris was able to convince detractors that properly written Forth is as maintainable as other high-level code and that its extensibility compensates adequately for its unfamiliarity to the programming establishment.

So how fast and how accurate are these programs?

Harris is particularly proud of his ground-tracking program. "We compared our results against DPTRAJ, which is JPL's big mainframe program for calculating orbits," he says. "And we found that over a twenty-four-hour period, our numbers varied from theirs by only one ten-thousandth of a degree in longitude and one one-hundredth of a second in time."

Calculating an orbit is exacting and complicated work. If the earth were a perfect sphere of uniform density, it could be treated as a point of mass for the purpose of gravitational calculations—that is to say, its gravitational effect upon an object in space would diminish in a smooth manner, exactly in proportion to the square of the object's distance from the center of the earth. In this hypothetical case, a satellite would trace a perfectly elliptical path, with the earth at one focus of the ellipse; its position could be predicted with complete accuracy by Kepler's laws of planetary motion.

The earth is only approximately a sphere, however, so the Keplerian model predicts only approximately the path that will be traveled by the space shuttle. Orbits of real objects around the real earth are subject to perturbations, caused by the nonuniform nature of the earth's gravitational field (and to a smaller degree the gravitational influence of objects other than the earth).

Harris's ground-tracking program and DPTRAJ work in rather different ways. The big machine adds the effects of disturbances over small intervals of time, while Harris's program uses averaging methods that predict effects over one orbit (perturbation theory). But the concordance between the two machines and two approaches is impressive: over a one-day period Harris's calculations of the shuttle's orbit differed from the mainframe's by only a ten-thousandth of a degree; his prediction of the shuttle's position along that orbit varied from DPTRAJ's by a mere hundredth of a second. That, as they say, is good enough for government work (and close enough for jazz).

Two years ago, for SIR-A, Harris performed some of the same kinds of computations using an Apple II. But SIR-B is a much more complex project than SIR-A, and the reliability and speed of the pc (particularly the speed afforded by the 8087) are proving invaluable. Exact awareness of orbital position and antenna angle are especially critical to this mission because the experimenters will be shooting pictures of certain targets from different locations in space on successive passes around the earth.

Who will use the information returned by SIR-B? Geologists, geographers, those involved in resource allocation planning—ultimately anyone from any number of disciplines whose concerns include the shape and substance of our planet.

The principal investigator will be Charles Elachi, who was also the principal investigator on SIR-A; coinvestigators have not been selected yet. "But of course," says Harris, "all this information will be made available to any interested scientist."

Including military scientists?

"Well, the military *is* interested, but they're not actively involved. They're sort of looking over our shoulder—particularly the Navy," Harris says.

If *you* are interested, you can avail yourself of the information as well. Findings from SIR-B will appear in JPL publications after the mission is completed. Write to the Jet Propulsion Laboratory in Pasadena, California, for details.

It will be a while before the publications are done, of course. In the meantime, preparations for the trip are still being made. The exact path of the shuttle and the exact targets to be imaged are still being selected.

And Harris is becoming something of a hero around JPL and throughout NASA—a champion of the small. "At first," he says, "people thought the kind of things I'm doing had to be done on mainframes. Now I get two or three calls a day from people wanting to talk about getting stuff off mainframes and putting them on a micro." ▲

You can count on 3M diskettes. Day after day.

Just like the sun, you can rely on 3M diskettes every day. At 3M, reliability is built into every diskette. We've been in the computer media business for over 30 years. And we've never settled in. We're constantly improving and perfecting our product line, from computer tape and data cartridges to floppy disks.

3M diskettes are made at 3M. That way, we have complete control over the entire manufacturing process. And you can have complete confidence in the reliability of every 3M diskette you buy.

Look in the Yellow Pages under Computer Supplies and Parts for the 3M distributor nearest you. In Canada, write 3M Canada, Inc., London, Ontario. If it's worth remembering, it's worth 3M diskettes.

3M hears you...

3M

Pick up a SixPakPlus™

Introducing . . . SixPakPlus™, the refreshing new 384KB multifunction card! In response to the changing needs of the IBM PC and PC-XT marketplace, AST Research, Inc. is proud to announce the latest addition to our line of multifunction enhancement products, the **SixPakPlus**! This new product is the result of extensive marketing research into the needs of IBM PC users whether they have the original 64K system board, the newer 256K system board, or the PC-XT. The **SixPakPlus** has been engineered to meet these needs at a competitive price while maintaining AST's high standards for quality and reliability.

The **SixPak**, as we like to call it, could have been named for the six banks of RAM on it. However, we like to think that it was named for the six functions of the card.

for your IBM® PC.

The features of the **SixPak** include:

1. RAM memory starting at 64K, user-expandable in 64K increments to 384K. This makes the **SixPak** ideal for the PC or PC-XT with a 256K system board; 384K on a **SixPak** added to 256K on the system board yields 640K, the maximum addressable user memory in these systems.

2. One Serial (async) communications port, configurable as either COM1 or COM2, for use with serial printers, modems, a "mouse," and other serial devices. The serial port has on-board jumpers for easy management of the RS-232C lines, simplifying the wiring of cables in many installations.

3. One Parallel (printer) port, configurable as LPT1 or LPT2 (LPT2 or LPT3 when the IBM monochrome card is installed), for use with the IBM/Epson and other compatible printers. The port is compatible with IBM diagnostics.

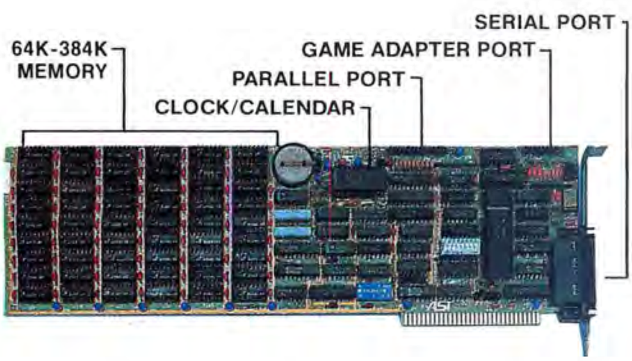
4. A Clock-Calendar with battery backup, featuring an easily replaceable Lithium battery and a quartz-controlled timebase for a high degree of accuracy.

5. An optional IBM-compatible Game Adapter port, for use with an IBM-type joystick. In conjunction with application programming, this game port may be used for cursor control, in generating graphics or for playing games at the end of your work day!

6. Every **SixPak** comes with an AST SuperPak utility diskette which includes SuperDrive and SuperSpool, the most powerful disk emulator and print spooler software you can get. These programs will greatly enhance the throughput of your PC or PC-XT by emulating disk drive and printer access at RAM speeds rather than the normal slower speed of mechanical devices. SuperPak is the first of such software to be compatible with both DOS 1.1 and DOS 2.0.

Most important of all, the **SixPak** comes with the AST "Plus," AST's unsurpassed reputation for quality, reliability, after-the-sale support, and overall design excellence that gives our products the best price/performance ratio in the industry! Hence the name, **SixPakPlus**!

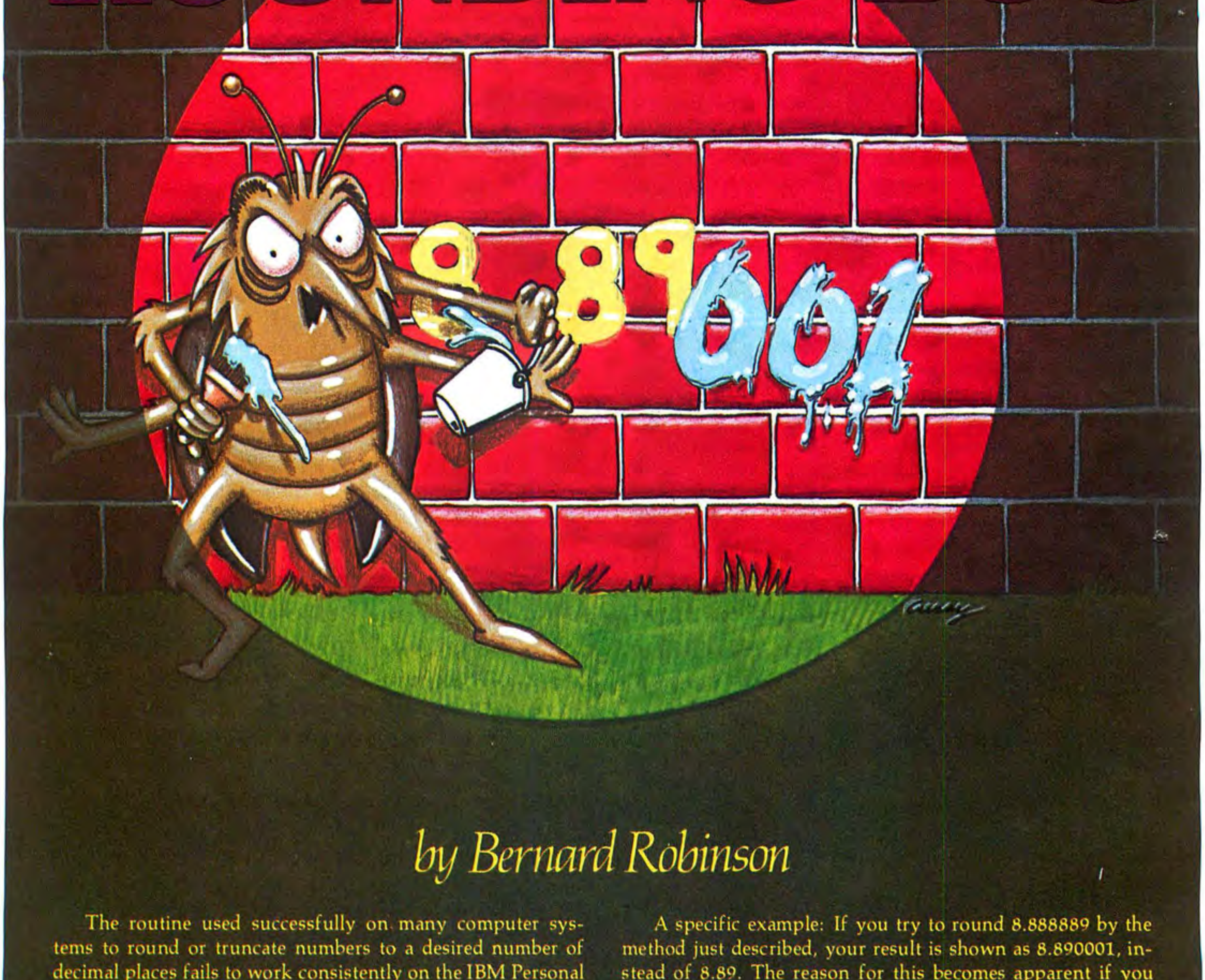
AST products are available from Computerland, Entre', ComputerMart, and selected dealers worldwide. Call factory if your dealer does not have the AST products you want.



® IBM is the registered trademark of International Business Machines

AST RESEARCH INC.
2372 Morse Avenue
Irvine, California 92714
(714) 540-1333 • TWX 295370ASTRUR
Dealer Inquiries Welcome

HOW TO GET 'ROUND THE ROUNDING BUG



by Bernard Robinson

The routine used successfully on many computer systems to round or truncate numbers to a desired number of decimal places fails to work consistently on the IBM Personal Computer.

The usual method (for rounding to two places) is to multiply the number to be rounded by 100, apply the *int* function to reduce the multiplied number to an integer, then divide by 100. For three places, multiply by 1,000, and so on. If you use this method with IBM Basic, you'll get the expected result most of the time, but occasionally you won't get the desired number of decimal places. (For a recent discussion of this matter, see "Questions and Answers," *Softalk*, June 1983.)

The method also fails when the *cint(x)* function is used and the value of x is greater than the largest single-precision integer handled by Basic.

A specific example: If you try to round 8.888889 by the method just described, your result is shown as 8.890001, instead of 8.89. The reason for this becomes apparent if you enter the following in direct mode:

```
X=8.89 : PRINT X
```

and the answer comes out 8.890001. You'll get the same error if you merely type *print 8.89* (and incidentally you can also behold these wonders on the Compaq, in Compaq's version of Microsoft Basic).

The fact is that Basic stores 8.89 as 8.890001. This represents an error of 1 in the seventh significant figure. The IBM Basic manual, on page 3-11, states, "With single-precision, seven digits are stored and up to seven digits are printed, although only six digits will be accurate." There it is, in Big Blue black and white.

Illustration by Mike Cressy

Applications Generator for the PC

The Only Programmable, 3-D Spreadsheet

EXEC, Page Dimension, and Ease of Use Make Report Manager the Professional's Choice

DATAMENSION CORPORATION
615 Academy Drive, Northbrook, IL 60062

Report Manager, the 3-D, programmable spreadsheet from DATAMENSION CORPORATION, is now available for the IBM/PC.

This breakthrough program brings the third dimension — pages — to report preparation. Its built-in language, EXEC, lets users generate a wide range of applications.

Report Manager has found wide acceptance among attorneys, engineers, ceo's, accountants, bankers, stock brokers and other professionals who demand the multi-dimensional power of a 3-D spreadsheet to prepare large, complex reports.

Report Manager lets the PC user put distinct spreadsheet pages into memory at one time, as if they were stacked one behind the other. All pages can be saved on diskette in a single file.

The program automatically expands to take advantage of all available RAM. If an application exceeds available memory, files can be linked, that is, the values of cells in separate files can depend on each other.

A single keystroke lets the user move from page to page. The PgDn key brings successive pages to the screen; PgUp brings previous pages into view.

All program commands, functions and formulas are active throughout all pages. Alterations to a cell on any page are reflected on all pages where that cell has been referenced.

Report Manager also lets PC users view their spreadsheet data base two additional ways. The column view brings a specified column from each page to the screen. The row view brings one row from each page to the screen.

Report Manager includes its own built-in programming language, EXEC. Files built under EXEC can load automatically when the PC power is turned on, then prompt naive users for data entry. As Report Manager's application generator, EXEC allows persons with no special training or knowledge to use Report Manager productively. Typical applications include order entry, invoicing and billing.

The charts below show how Report Manager satisfies the professional's demand for multi-dimensional reporting power.

	3-D spreadsheet	Max. no. columns/file	Max. no. rows/file	Max. no. separate pages/file	Max. no. of cells ($\times 10^6$)	View crossection of cols.	View crossection of rows	Use up to 640K RAM	Built-in programming language	No. of program statements	Files request & receive data	Keystroke-playback files	Make your own menus	Chaining of files possible	Self-starting files	Auto. center of vert. headings	Auto. center of horiz. headings	No. graphing characters used	Printer configuration tables	Replicate (copy) option	Replicate in 3 dimensions	Replicate only cell format	Replicate only cell contents
Report Manager™	**Y**	**255**	**255**	**255**	**16.0**	**Y**	**Y**	**Y**	**Y**	**17**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**200+**	**Y**	**Y**	**Y**	**Y**	**Y**
Multiplan™	N	63	255	1	0.02	N	N	N	N	0	N	N	N	N	N	N	N	1	N	Y	N	N	N
1-2-3™	N	256	2048	1	0.50	N	N	N	Y	4	Y	N	Y	N	N	N	N	3	N	Y	N	N	N
VisiCalc™	N	63	254	1	0.02	N	N	N	N	0	N	N	N	N	N	N	N	1	N	Y	N	Y	Y
SuperCalc™	N	63	254	1	0.02	N	N	N	N	0	N	N	N	N	N	N	N	1	N	Y	N	N	N

	Report linking (consolidation)	Top-of-column totals	On-line reference guide	Choose on/off coord. grid	Display formulas or their results	Repeat cell contents	Name cells in English	Write formulas in English	Variable, indiv. col. widths	Choose no. of decimal places	Titles available	Windows available	Choose auto. recalculation	Supports color or monochrome	XT compatible	Store reports in ASCII format	Read DIF™ files	Data link with Manager Series™	Copyable program included	Printed program tutorial	Data disk with examples	Usable templates included	Telephone support
Report Manager™	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
Multiplan™	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	DNA	N	Y	Y	Y	Y
1-2-3™	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	DNA	N	Y	Y	N	Y
VisiCalc™	N	N	N	N	N	Y	N	N	Y	Y	Y	Y	Y	Y	N	Y	Y	DNA	N	Y	N	N	Y
SuperCalc™	N	N	Y	Y	Y	N	N	N	Y	N	Y	Y	Y	Y	Y	Y	N	DNA	Y	Y	N	Y	Y

	Preprogrammed functions	No. of arithmetic operators	No. of relational operators	No. of trig. functions	Date + time arithmetic	Date interval in days	Day of week function	Linear regression	Look-up in 3 dimensions	Mode function	Time interval in seconds	Standard deviation	Trend line analysis	Radian/degree conversion	Amortization schedule	Effective interest rate	Future value for annuity due	Loan payment	Number of payments	Payment for annuity due	Present value	Present value for annuity due	Present value for a bond
Report Manager™	**57**	**6**	**6**	**6**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**	**Y**
Multiplan™	41	6	6	4	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	N	N	N	N
1-2-3™	51	5	6	7	N	N	N	N	N	N	N	Y	N	N	N	N	N	Y	N	N	Y	N	N
VisiCalc™	31	5	6	6	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
SuperCalc™	36	5	6	6	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N

Basic is well within this stated accuracy when it stores 8.89 as 8.890001. But that's no help to you when you want to round 8.8888889 to 8.89.

What should you do? The answer depends on what you're out to accomplish. It may be that your objective is to get a neat-looking printout; if that's the case, the best way is probably to use *print using* or *lprint using*, as IBM has recommended all along, ever since David Walonick spilled the bad news to the *New York Times*. But what if you have to use that rounded number in further calculations? The output statements are of no help then.

What follows is a recommended solution. A number of other possible solutions were tried before this one was arrived at. For example, if you do the following:

```
A$="8.89" : PRINT VAL(A$)
```

Basic prints 8.89, as you would want. That's fine, but if you then do a:

```
X=VAL(A$)
```

you're back to 8.890001.

The recommended approach uses double precision. This makes sense, since if 8.890001 is not close enough to 8.89, then one is probably going to have to use double precision.

The *cint(x)* function, which rounds the fractional part of a number when it converts a real to an integer, does not function on double-precision numbers. The *int(x)* and *fix(x)* functions do work on double-precision numbers. Note the difference between these two functions: *int(x)* always goes to the next lower integer (it's sometimes called a "floor" function), while *fix(x)* drops the fractional part of the number and retains the integer part unchanged. *Int(x)* is more suitable for conventional rounding.

The recommended solution is to use the following user-defined function:

```
DEF FNDRND#(G#,K)=INT((G#*10^K)+.5)/10^K
```

CAMEO

HARD DISK SYSTEMS ARE THE SOLUTION TO:

- **Reliable Back-Up Requirements**
- **Large Data Storage Needs**

CAMEO large capacity high performance hard disk systems are the ideal mass storage system for use in those business environments where a company's data bank itself is understood to be one of the most valuable parts of the whole micro-computerized office system. The data bank was costly to acquire and could be more costly to replace. Regular access to it is needed every day without fail.

CAMEO'S subsystem 1050, consists of 40 formatted megabytes of on-line winchester data storage and 10 formatted megabytes of removable winchester cartridge back-up capability. In the unlikely event of a problem with the fixed winchester drive the removable cartridge drive can perform as a fully operable full function high performance on line random access winchester hard disk system. It provides highly reliable (full CRC error checking on full spec winchester media) permanent off-line data storage capability for permanent data archiving.

CAMEO'S configuration options allow multiple computers to access all areas of the disk as well as virtual RAM disk configuration from add-on memory boards. Higher capacity systems and nationwide third party maintenance are available. Immediate availability through CAMEO'S regional distributor network.

IBM VICTOR
ZENITH NEC
COMPAQ EAGLE

A CAMEO interface is available for the above as well as almost any brand of high performance micro computer you are considering for your office use.

Contact your local CAMEO dealer or:

- **CAMEO ELECTRONICS, INC.**
  1626 Clementine
  Anaheim, Ca. 92802
  (714) 535-1682

- **CAMEO EUROPE**
  Postfach 227
  8120 Weilheim
  West Germany
  TEL 08802-691
  TLX 59903 CAMEO D

where G# is the dummy variable standing for the double-precision input number to be rounded and K is the desired number of decimal places. The "+.5" in this expression gives rounding such that fractional parts of .5 or more are rounded up to the next greater number, positive or negative.

If many numbers are rounded this way, a slight bias is introduced, since fractional parts of exactly .5 are always rounded up. Numerical analysts recommend a rounding scheme such that when the dropped fraction is exactly .5, the rounding goes to the nearest even number. The function defined here can be modified to supply this kind of rounding. Here's the way to round numbers ending in .5 to the nearest even number:

```
DEF FNERND#(G#,K)=INT(((G#/2)*10^K)+.5)*2/10^K
```

This function is adapted from a routine for pocket calculators found in *Calculator Tips and Routines*, by John Dearing (published by Corvalis Software).

To use either of these methods, place the selected *def fn* statement in the program ahead of where you call it the first time. Say the variable you want to round is X#, you want the rounded number stored in variable X1#, and you want to round to three places. Here's your statement:

```
X1#=FNERBND#(X#,3)
```

Listing 1 shows the use of the *def fndrnd( )* method in a program that converts Fahrenheit temperatures to Celsius. For comparison, the "old way"—using single precision and *cint ( )*—is shown alongside the new method. A few values known to cause trouble in the old method are included. The double-precision value of the Celsius temperature is printed out for comparison.

Listing 2 includes both *def fndrnd( )* and *def fnernd( )* and compares the outputs obtained for a few values, using each of them. Note that the use of double precision allows rather large numbers to be handled correctly by the rounding routine. ▲

```
06-06-1983     16:16:07
10 DEF FNDRND#(G#,K)=INT((G#*10^K)+.5)/10^K
40 DATA 13,15,16,48,49,152,190,199
50 LPRINT "F","D#                  ","OLD WAY","NEW WAY"
60 K=2
70 FOR I=1 TO 8
80 READ FX
82 F#=FX
90 D#=(5*(F#-32))/9
92 DX=(5*(FX-32))/9
100 DX1=CINT(DX*10^K)/10^K
110 D1#=FNDRND#(D#,K)
120 LPRINT F#,D#,DX1,D1#
130 NEXT
140 END
```

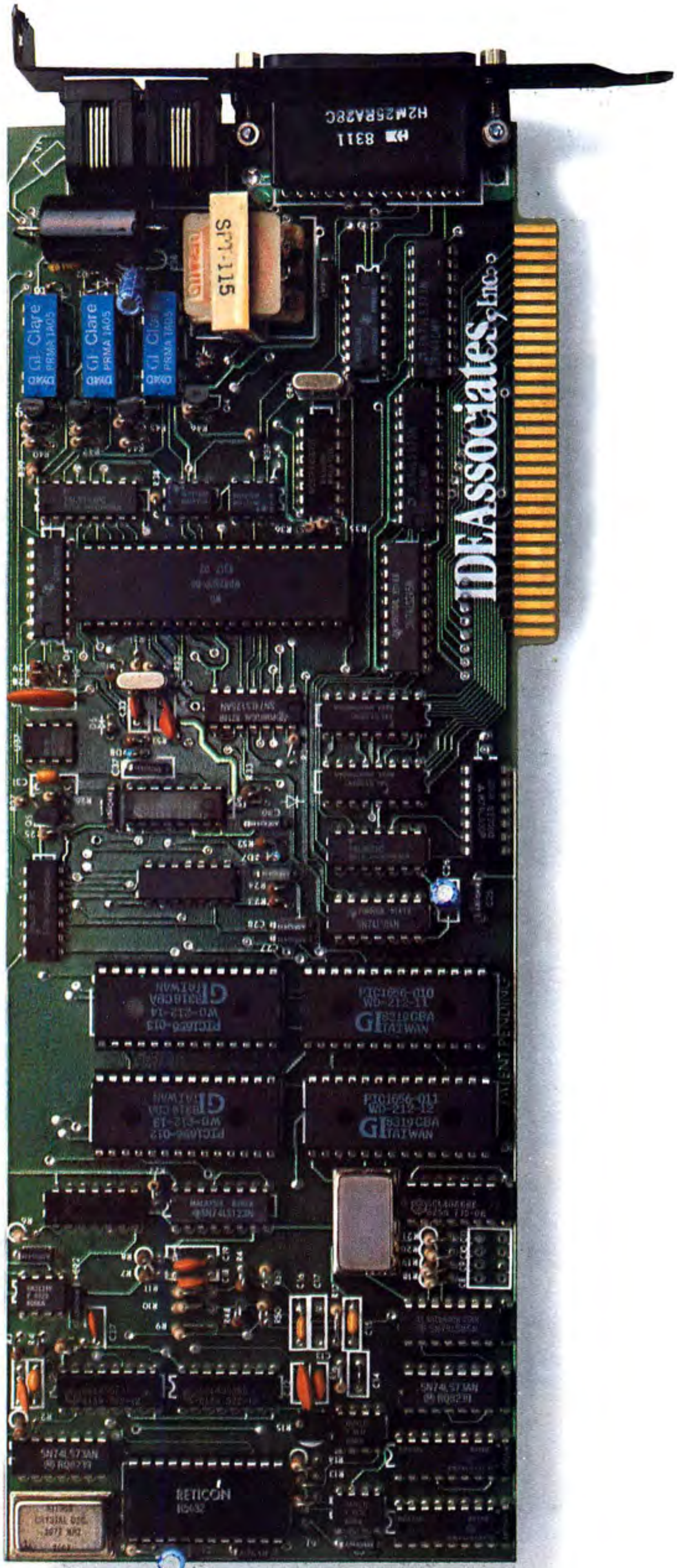
F	D#	OLD WAY	NEW WAY
13	−10.55555555555556	−10.56	−10.56
15	−9.444444444444445	−9.439999	−9.44
16	−8.888888888888889	−8.890001	−8.89
48	8.888888888888889	8.890001	8.89
49	9.444444444444445	9.439999	9.44
152	66.66666666666667	66.67	66.67
190	87.77777777777778	87.78	87.78
199	92.77777777777778	92.78	92.78

Listing 1.

```
06-06-1983     16:12:30
10 DEF FNDRND#(G#,K)=INT((G#*10^K)+.5)/10^K
20 DEF FNERND#(G#,K)=INT(((G#/2)
   *10^K)+.5)*2/10^K
30 DATA 123456789.12355#,100027823.5465#,
   137873545.6755#,167290001.44444445#
40 DATA 734898922.5055#
50 LPRINT "GIVEN NUMBER      ","FNDRND( )
","FNERND( )"
60 K=3
70 FOR I=1 TO 5
80 READ D#
100 D1#=FNDRND#(D#,K)
110 D2#=FNERND#(D#,K)
120 LPRINT D#,D1#,D2#
130 NEXT
140 END
```

GIVEN NUMBER	FNDRND( )	FNERND( )
123456789.12355	123456789.124	123456789.124
100027823.5465	100027823.547	100027823.546
137873545.6755	137873545.676	137873545.676
167290001.4444444	167290001.444	167290001.444
734898922.5055	734898922.506	734898922.506

Listing 2.



Introduce your IBM-PC to a fast-talking card sharp.

The IDEAComm 1200: Fast, reliable communications on a plug-in card.

Now you don't need an external modem to get 1200 baud communications capability for your IBM-PC. The IDEAComm 1200 is an integral modem that combines the functions of an in-board asynchronous communications card and an out-board modem in one integrated plug-in unit. Snap it into an expansion slot, plug in a modular telephone jack, and you have direct 1200 *or* 300 baud communications without the confusion and added expense of external boxes and cables. And there are no switches to set; the speed is software-selectable or automatically selected by the card. Plus, we've designed in outstanding reliability: four powerful on-board microprocessors replace conventional analog circuitry with digital precision and reliability.

One Card That's a Great Deal

The fast-talking IDEAComm 1200 is one sharp card: it eliminates the need for an external modem, gives you 1200/300 baud communications capability with the exceptional reliability of digital design; provides an additional plug so you don't lose regular telephone voice communications; and includes a standard RS232C interface and connector that can be used as an additional serial port. All for only $545. That's about the cost of a conventional external modem alone, saving you the entire cost of the internal asynchronous card.

One Card Gives You a Full Hand

The IDEAComm 1200 comes complete with software diskette enabling easy, menu-driven selection of all communications, autodial, and auto log-on procedures, and easy recall at the touch of a single key. It also allows the IDEAComm 1200 to work with most other terminal emulation packages—including IBM Comm 2.0 communications software.

Technical Specifications

Speed 0-300 bps or 1200 bps **Data Format** Serial, binary, 7-8 data bits, 1-2 stop bits **Dialing Capability** Touch-Tone (TM) or rotary dial pulse **Operation** Full Duplex	**IBM Port** COM 1 **Modem Compatibility** Bell System 103 or 212A in both Originate and Answer mode **Physical Data** 3.9" x 10.8" x .6" (one card slot)

IDE Software

We've Stacked Our Deck With Better Ideas

IDE's Better Ideas for your IBM-PC include expansion memory boards, "The IDEABoard" combination boards, fixed or removable Winchester disk drives for either internal or external mounting, and printer spooler and RAMFloppy disk emulation software. We have Better Ideas on product support, too. Superb technical support and documentation, including full programming specifications for the IDEAComm 1200 . . . an aggressive pricing policy . . . a unique upgrade policy that lets you trade up any IDE product for another in our line . . . and our full year warranty.


Expansion Memory Board


"The IDEAboard"

Get the IDEA

To see IDE's Better Ideas for yourself, contact any of the knowledgeable dealers listed below. If there is no listing for a dealer near you, call us. We'll provide you with the name of your nearest dealer or take your order over the phone.

1-800-257-5027

(In Massachusetts, call 617-275-4430)


Removable Winchester disk drive

IDE Associates
Better ideas for personal computers.

IDEAssociates, Inc. 7 Oak Park Drive, Bedford, MA 01730

IDE Dealers: ☐ Alaska: VFM Electronics, Fairbanks ☐ Arizona: Abbot Data Products, Glendale; Computer Emporium, Tucson ☐ California: Q-Bits, Mountain View; Stoneware Ltd., La Jolla ☐ Connecticut: The Computer Establishment, Old Saybrook; MicroAge, Danbury; American Business Computers, Groton; Creative Systems, E. Granby ☐ Hawaii: Peripherals Hawaii, Kahuku ☐ Illinois: DM Micro Systems, Collinsville ☐ Iowa: ComputerLand, Des Moines ☐ Kansas: Computype, Manhattan ☐ Massachusetts: Arby Corporation, Cambridge; TEK Microsystems, Boston ☐ Michigan: The Computer Workshop, Houghton; TransNational Corp., Troy; Computer Rentals, Royal Oak ☐ Minnesota: Computer Depot/Daytons Computer Centers, Minneapolis, St. Paul, Burnsville Center, Ridgedale, Rosedale, Southdale ☐ Nebraska: ComputerLand, Lincoln; ComputerLand, Omaha ☐ New York: Future Data, NYC; Oppenheimer Software, NYC; Yankee Computer Services, Putnam Valley ☐ North Carolina: Surveyor's Supply, Apex; Computer South, Charlotte; Microcomputer Sales, Fayetteville; The Computer Exchange, Durham ☐ Ohio: U-Compute, Toledo ☐ Oklahoma: Advanced Financial Technology, Oklahoma City; Biologic Instruments, Stillwater ☐ Pennsylvania: Software Projections, Inc., Flourtown ☐ South Carolina: Berry's Office Automation, Lugoff ☐ South Dakota: Computer Terminal, Sioux Falls ☐ Tennessee: FICON, Nashville ☐ Texas: MicroSpec, Plano; Video Station, Plano; Southwest Automation, Houston ☐ Utah: Weidner Communications Corp., Provo ☐ Virginia: Micro Solutions, Reston ☐ Washington: Telecalc, Seattle

International Distributors: Austria, Switzerland, West Germany: Computer 2000 GmbH, Munich ☐ United Kingdom: KPG Hardware House, London ☐ France: Micro Connection International, Paris ☐ Israel: O.D.C. Systems, Natanya

DOS from Pascal: Getting There the Easy Way

by Edward Mitchell

PC-DOS, the IBM Personal Computer Disk Operating System, provides a wealth of functions for use by application programs. Generally, high-level language programs have had to access DOS by using individual assembly language routines for each function.

But there is a better way: Write a single assembly language subroutine and code each of the needed DOS functions directly in Pascal by calling the DOS subroutine.

This article describes a simple implementation of the DOS procedure and how it can be used to do DOS input/output and file operations. The reader is expected to be familiar with IBM Pascal. And a knowledge of the IBM's 8088 assembly language and DOS operating system would help, but it's not absolutely necessary.

The Conventional Approach. To illustrate the usual method of accessing DOS, let's look at a DOS function that reads a single character from the keyboard, *without* printing the entered key on the display.

To perform this function, the program calls the Disk Operating System, which in effect calls a subroutine deep inside DOS to actually read the character. In assembly language, the read-character-without-echo subroutine is called by writing,

```
MOV   AH,8
INT   21H
```

This code loads the read character function number (here, 8) into the 8088 AH register and performs interrupt 21H, which calls DOS.

DOS waits for a key to be struck, returning the key's ASCII code in register AL. Listing 1 shows an assembly language routine called READKEY and a Pascal program that calls READKEY, demonstrating a call to DOS. The IBM DOS manual, Appendix D, describes the format for each of the DOS functions.

A Better Way. Since the code for each DOS call is essentially the same, except for the values placed in the registers, a better way is to create a procedure called DOSCALL( ) and pass to it the register values.

The 8088 has four sixteen-bit general-purpose registers: AX, BX, CX, and DX. Each of these can be viewed as two eight-bit registers, (AL, AH), (BL, BH), (CL, CH), and (DL, DH). To assign values to the registers as either sixteen-bit or eight-bit values, we need to create a Pascal case-variant record.

```
Type Register = Record
                  Case Boolean Of
                     TRUE  : ( L : BYTE;
                               H : BYTE) ;
                     FALSE : ( X : WORD );
                  End;

Var
      A, B, C, D : Register;
```

The case-variant record type allows each of the variables to be referenced three different ways: A.X to get all sixteen bits, A.L for the low byte, and A.H for the high byte.

Now, by creating a DOSCALL () procedure (see listing 2), we can pass the register value to and from the assembly language interface. Listing 3 shows how A.H is set to 8, DOSCALL() is called, and on return A.L holds the entered key value.

The input of a single typed key is fairly easy. Other functions, such as buffered line input, require that the DX register contain the *memory address* of the buffer's location. IBM Pascal provides the prefix operator ADR to return the address of a variable. So, to do buffered input, the Pascal program sets D.X to the address of a buffer variable.

According to the IBM DOS manual, function 'A'H (hexadecimal digit 'A') performs buffered input. The buffer that DX points to should be preformatted so that byte 1 contains the length of the buffer in bytes. On return, byte 2 will be set to the actual number of characters typed, and byte 3 out to the end of the buffer will be filled with the entered text. Listing 5 shows a procedure that calls DOS to perform this function.

File Operations. The parse filename function ('29'H) uses the 8088's SI, DI, and ES registers. Rather than add these additional registers into the DOSCALL procedures (SI, DI, and ES are rarely used in the DOS functions), a second, extended DOSCALL procedure, defined as

```
PROCEDURE XDOSCALL (Var A, B, C, D, SI, DI : Register);
```

is created. Internally, ES is set equal to the DS register so that XDOSCALL can access Pascal variables. XDOSCALL is shown in Listing 4. By separating the two procedures, programs avoid wasting time passing the values of the extra registers around.

To use file functions, a *file control block* (or FCB) containing information about each file must be defined. The file FILKQQ.INC on the Pascal PAS1 disk contains the FCB definition (also see listing 6).

Listing 6 shows how XDOSCALL() and DOSCALL() are used to create four functions, PARSEFN, FILEOPEN, FILEREAD, and FILECLOSE.

Summary. The simple DOSCALL() interface enables IBM Pascal programs to access all of DOS's features directly from Pascal. Programs can now do directory listings, determine the current default disk setting, set up control-break handlers, and perform random block reads and writes—all without resorting to assembly language. Specific DOS calls are described in detail in the IBM DOS manual.

References

"Appendix D: DOS Interrupts and Function Calls," *IBM DOS Reference Guide.*

IBM Pascal Reference Guide.

IBM Macro Assembly Reference Guide.

McCune

...audio cassettes by mail

The best of PC'83

SAN FRANCISCO
Friday–Sunday, June 17–19, 1983
Brooks Hall/Civic Center

Selected Conference Programs available now . . .

Individual sessions $8 each
Any six sessions $40
with cassette album
Complete program $130
with cassette albums

- ☐ 1. Linking PC's to Large Host Networks/Morel
- ☐ 2. Venture Capital and the Computer Industry/Gold, Marzak, Gianos
- ☐ 3. A Printer's a Printer, Isn't It?/Dean
- ☐ 4. Developing Vertical Market Software/Finkler
- ☐ 5. Software Market Trends/Stockett
- ☐ 6. How To Cut Through Computer Jargon/Dublin
- ☐ 7. PC's in the Corporate Suite/Posner
- ☐ 8. Voice Response Systems/Pelton
- ☐ 9. What to Expect From Portable Computers/Edens
- ☐ 10. Overcoming Executive Computerphobia/Orton
- ☐ 11. The Outraged Consumer's Guide to Word Processors/Dean
- ☐ 12. A Technical Guide for Non-Technical Beginners/Magid
- ☐ 13. Introduction to Programming in PASCAL
- ☐ 14. PC Operating Systems — Panel
- ☐ 15. PC's in Large Organizations/Begun
- ☐ 16. How to Select and Interface Disks to the PC/Moulton
- ☐ 17. Choosing a Computer Retailer/Dean
- ☐ 18. Interface Boards and Peripherals/Moulton
- ☐ 19. The Battle of the IBM Disk Operating System and Integrated Software/Battle
- ☐ 20. Office Automation

Cassette Tape Order Form

Please add $5.00 to all mail orders for postage and handling.
Please make checks payable to:

McCune Audio/Visual
951 Howard Street
San Francisco, CA 94103

Cassette total: $ ______________
Tax: ______________
Shipping: $5.00
TOTAL: ______________

(Add 6% state tax if mailed within Calif. and 6.5% if within BART district.)

Name ______________________ Phone ______________

Address ______________________________________

City ______________ State ______________ Zip ______________

Paid by: ☐ CHECK ☐ MONEY ORDER

Note: Mail orders by check or money orders only

DOS from Pascal

```
Assembly    segment 'code'
dgroup      group   data
assume      cs:Assembly, ds:dgroup
assume      es:dgroup, ss:dgroup

    public  readkey

; FUNCTION ReadKey : Integer;
readkey             proc far
            push    bp
            mov     bp,sp
            mov     ah, 8       ; DOS Input, No Echo
            int     21h
            xor     ah,ah       ; Clear Ah register
                                ; Return character in AX
            pop     bp
            ret
readkey             endp

Assembly            ends
                    END
```

Listing 1(a).

```
Program DemoRead (Input, Output);

Var

        Ch : Char;

Function ReadKey : Integer; Extern;

Begin
  Repeat
    Ch := Chr(ReadKey);
    Writeln ('You pressed:  ',Ch);
  Until  Ch='*';
End.
```

Listing 1(b).

```
Dos             segment 'code'
dgroup          group   data
assume          cs:Dos, ds:dgroup
assume          es:dgroup, ss:dgroup

        public  DosCall

DosCall Proc Far

; Procedure DosCall
;           ( VAR A, B, C, D: Register);
D_Offset        EQU     6
C_Offset        EQU     8
B_Offset        EQU     10
A_Offset        EQU     12

        Push    BP
        Mov     BP,     SP
        Mov     AX,     DS
        Mov     ES,     AX
        Mov     DI,     [BP+D_Offset]
        Mov     DX,     [DI]
        Mov     DI,     [BP+C_Offset]
        Mov     CX,     [DI]
        Mov     DI,     [BP+B_Offset]
        Mov     BX,     [DI]
        Mov     DI,     [BP+A_Offset]
        Mov     AX,     [DI]
        Int     21H
        Mov     DI,     [BP+A_Offset]
        Mov     [DI],     AX
        Mov     DI,     [BP+B_Offset]
        Mov     [DI],     BX
        Mov     DI,     [BP+C_Offset]
        Mov     [DI],     CX
        Mov     DI,     [BP+D_Offset]
        Mov     [DI],     DX
        Pop     BP
        RET     8

DosCall EndP

Dos     EndS
        End
```

Listing 2.

```
Program ReadDemo ( Input, Output);

Type
  Register =
    Record
      Case Boolean Of
        True  :       ( L  : Byte   ;
                        H  : Byte    );
        False :       ( X  : Word    );
      End;

Var
      A, B, C, D  : Register;
      Ch          : Char;

Procedure DosCall
        ( Var A, B, C, D : Register );
        Extern;

Begin

  Repeat
    A.H := 8;
    DosCall (A, B, C, D);
    Ch := Chr(A.L);
    Writeln('You pressed:  ',Ch);
  Until  Ch='*';
End.
```

Listing 3.

```
XDos            segment 'code'
dgroup          group   data
assume          cs:XDos, ds:dgroup
assume          es:dgroup, ss:dgroup

        public  XDosCall

XDosCall Proc Far

; Procedure XDosCall
;   ( VAR A, B, C, D, SI, DI: Register);
DI_Offset       EQU     6
SI_Offset       EQU     8
D_Offset        EQU     10
C_Offset        EQU     12
B_Offset        EQU     14
A_Offset        EQU     16

Push    BP
Mov     BP,     SP
Mov     AX,     DS
Mov     ES,     AX
Mov     DI,     [BP+D_Offset]
Mov     DX,     [DI]
Mov     DI,     [BP+C_Offset]
Mov     CX,     [DI]
Mov     DI,     [BP+B_Offset]
Mov     BX,     [DI]
Mov     DI,     [BP+A_Offset]
        Mov     AX,     [DI]
        Mov     DI,     [BP+SI_Offset]
        Mov     SI,     [DI]
        Mov     DI,     [BP+DI_Offset]
        Mov     DI,     [DI]
        Int     21H
        Push    DI              ; Save for a moment
        Mov     DI,     [BP+A_Offset]
        Mov     [DI],     AX
        Mov     DI,     [BP+B_Offset]
        Mov     [DI],     BX
        Mov     DI,     [BP+C_Offset]
        Mov     [DI],     CX
        Mov     DI,     [BP+D_Offset]
        Mov     [DI],     DX
        Mov     DI,     [BP+SI_Offset]
        Mov     [DI],     SI
        Pop     AX              ; Restore old DI
        Mov     DI,     [BP+DI_Offset]
        Mov     [DI],     AX
        Pop     BP
        Ret     12

XDosCall EndP

XDos    EndS
        End
```

Listing 4.

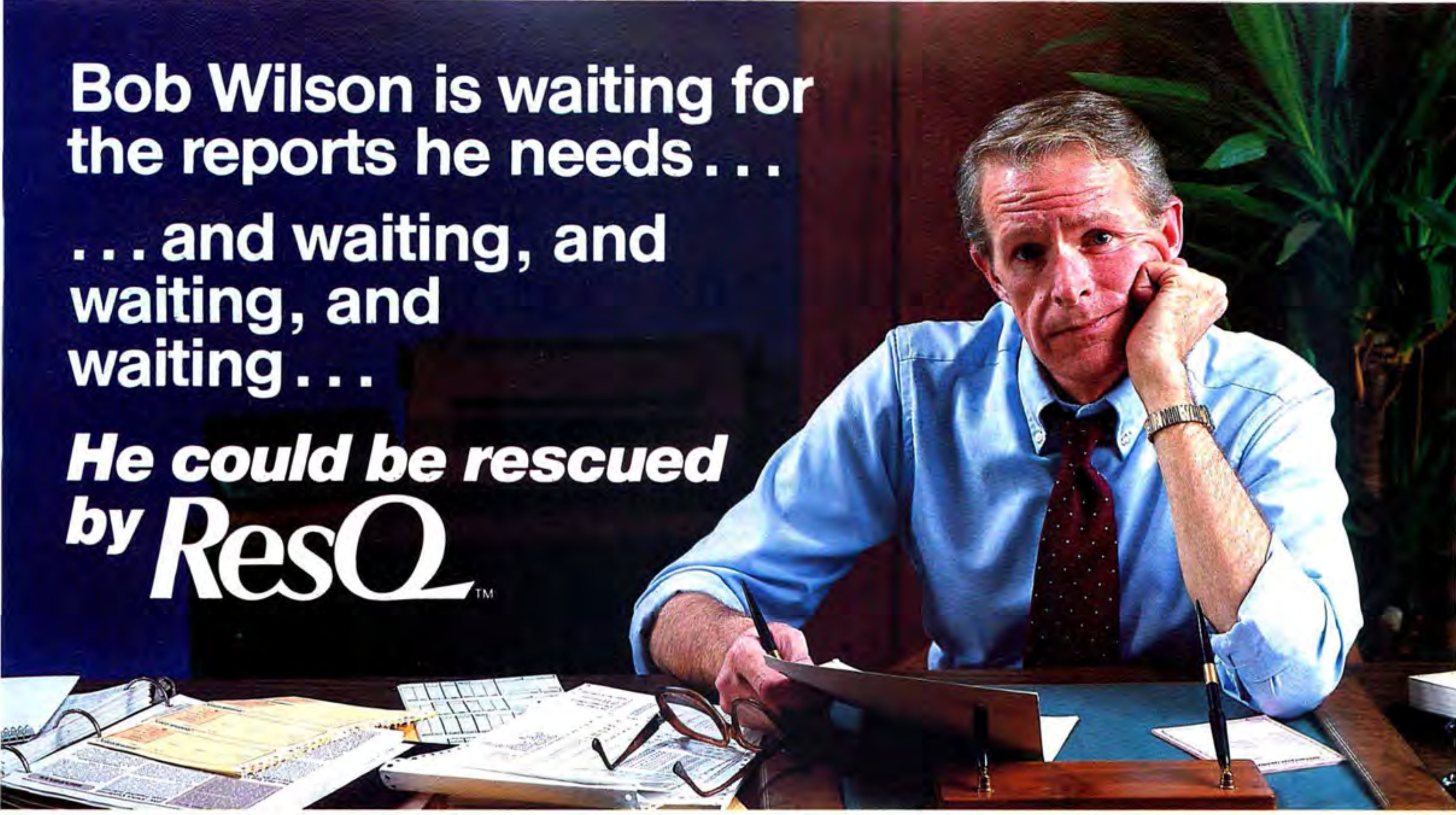
```
Program InputDemo (Input, Output);

 Type
   Register =
     Record
       Case Boolean Of
         True  :      ( L  : Byte   ;
                        H  : Byte   );
         False :      ( X  : Word   );
       End;
 Var
     Buffer : String(128);
     A, B, C, D : Register;
     I         : Integer;
     Length : Integer;

   Procedure DosCall
     ( Var A, B, C, D : Register );
     Extern;

   Begin
     Write('Enter some text? ');
     Buffer[1] := Chr(126);
     D.X := Wrd ( Adr Buffer);
     A.H := 16#0A;
     DosCall (A, B, C, D);
     Writeln('You entered:');
     Length := Ord(Buffer[2]);
     For I := 3 To 3+Length Do
       Write(Buffer[i]);
     Writeln(Null);
   End.
```

Listing 5.

```
(*$DEBUG-*)
(*$WARN-*)

Program FileOps ( Input, Output );

Const
      BUFSIZE            = 512;   (* Disk Buffer/Record Size *)

   (* Error Codes returned by DOS File Routines *)
   BADDRIVE         = 1;  (*Bad Drive name in filename*)
   WILDFILENAME     = 2;  (* Filename contains wildcard chars *)
   FILENOTFOUND     = 3;  (* File isn't on the disk *)
   DIRECTORYFULL    = 4;  (* Disk Directory has too many entries *)
   ENDOFFILE        = 5;  (*End Of File occurred on read*)
   PARTIALREAD      = 6;  (* End of file occurred on read *)
   DISKFULL         = 7;  (* No more room on the disk *)
   DISKCHANGED      = 8;  (* Disk changed with file still open *)

Type

      Register =
        Record
          Case Boolean Of
            True  :      ( L  : Byte   ;
                           H  : Byte   );
            False :      ( X  : Word   );
          End;

      FCB = Record
   Drive     [ 0]  : BYTE;              (* Drive Number *)
   FName     [ 1]  : STRING (8);        (* File name *)
   Ext       [ 9]  : STRING (3);        (* File name extension *)
   CurBlkN   [12]  : WORD;              (* Current Block Number *)
   RecSize   [14]  : WORD;              (* Record Size *)
   FSizeLo   [16]  : WORD;              (*Low Word of File Size*)
   FSizeHi   [18]  : WORD;              (* High Word of File Size *)
   DateF     [20]  : WORD;              (* Date Field *)
   Resrvd    [22]  : STRING    (10);    (* DOS Reserved Space *))
   CurSect   [32]  : BYTE;              (* Current Sector *)
   RecNum    [33]  : WORD;              (* Record Number, Low Word *)
   RecNum3   [35]  : BYTE;              (* Byte 3 of record number *)
   RecNum4   [36]  : BYTE;              (* Byte 4 of record number *)
 End;

 BufType =
   Packed Array [1..BUFSIZE] Of Char;

 Var
      IoError      : Integer;
      Buffer       : BufType;
      RecNum       : Integer;
      FName        : Lstring(20);
      Cmd          : Lstring(10);
      I            : Integer;
      F            : FCB;

Procedure DosCall (Var A, B, C, D : Register ); Extern;
Procedure XDosCall ( Var A, B, C, D, SI, DI : Register );
Extern;
Procedure ParseFN ( Var F : FCB; Var FN : LString );
Var
      A, B, C, D, SI, DI : Register;

Begin
  Fn.Len := Fn.Len + 1;
  Fn[Fn.Len] := ' '; (* Append terminator *)
  SI.X := Wrd ( Adr Fn ) + 1;
  DI.X := Wrd ( Adr F );
  A.H := 16#29; (* DOS Parse Filename Func. *)
  A.L := 1;       (* Parse request code *)
  XDosCall (A, B, C, D, SI, DI);
  If A.L=1 Then
    IoError := WILDFILENAME
  Else
  If A.L(16#FF Then
    IoError := BADDRIVE
  Else
    IoError := 0;
End;

Procedure FileOpen ( Var F : FCB;
          Var FN : LString; SizeRecord : Integer );
Var
      A, B, C, D : Register;
Begin
  ParseFN ( F, FN );
  If IoError <> 0 Then
    Return;
  D.X := Wrd ( Adr F );
  A.H := 16#F;  (* DOS Open Existing File Code *)
  DosCall ( A, B, C, D );
  If A.L = 16#FF Then
    IoError := FILENOTFOUND
  Else
    IoError := 0;
  F.RecSize := SizeRecord;
  F.CurBlkN := 0;
  F.CurSect := 0;
End;
```

```
Procedure FileCreate ( Var F : FCB;
          Var FN : LString; SizeRecord : Integer );
Var
        A, B, C, D : Register;
Begin
  ParseFN ( F, FN );
  If IoError <> 0 Then
    Return;
  D.X := Wrd ( Adr F );
  A.H := 16#16; (* DOS Create File Code *)
  DosCall ( A, B, C, D );

  If A.L = 16#FF Then
    IoError := DIRECTORYFULL
  Else
    IoError := 0;
  F.RecSize := SizeRecord;
  F.CurBlkN := 0;
  F.CurSect := 0;
End;

Procedure FileClose ( Var F : FCB );
Var
        A, B, C, D : Register;
Begin
  D.X := Wrd ( Adr F );
  A.H := 16#10; (* DOS Close File Code *)
  DosCall (A, B, C, D);
  If A.L=16#FF Then
    IoError := DISKCHANGED
  Else
    IoError := 0;
End;

Procedure SetDTA ( BufAdr : WORD );
Var
        A, B, C, D : Register;
Begin
  A.H := 16#1A; (* Set Disk Transfer Address *)
  D.X := BufAdr;
  DosCall ( A, B, C, D );
End;

Procedure FileRead ( Var F : FCB;
          Var Buffer : BufType; RecNumber : Integer );
Var
        A, B, C, D : Register;
Begin
  SetDTA ( Wrd( Adr Buffer ) );
  F.RecNum := RecNumber;
  F.RecNum3 := 0;
  F.RecNum4 := 0;
  D.X := Wrd ( Adr F );
  A.H := 16#21; (* DOS Random Read Code *)
  DosCall (A, B, C, D );
  If A.L = 1 Then
    IoError := ENDOFFILE
  Else
  If A.L = 3 Then
    IoError := PARTIALREAD
  Else
    IoError := 0;
  End;

Procedure FileWrite ( Var F : FCB;
          Var Buffer : BufType; RecNumber : Integer);
Var
        A, B, C, D : Register;
Begin
  SetDTA ( Wrd ( Adr Buffer ));
  F.RecNum := RecNumber;
  F.RecNum3 := 0;
  F.RecNum4 := 0;
  D.X := Wrd ( Adr F );
  A.H := 16#22; (* DOS Random Write Code *)
  DosCall (A, B, C, D);
  If A.L = 1 Then
    IoError := DISKFULL
  Else
    IoError := 0;
End;

Begin
  Writeln('Demonstration of DOS File Calls');
  Repeat
    Write('C)reate O)pen R)ead W)rite K)lose Q)uit ? ');
    Readln(Cmd);
    If Cmd = 'C' Then
    Begin
      Write('Enter Filename: ');
      Readln(FName);
      FileCreate (F, FName, BUFSIZE);
    End
    Else
    IF Cmd = 'O' Then
    Begin
      Write('Enter Filename: ');
      Readln(Fname);
      FileOpen (F, FName, BUFSIZE);
    End
    Else
    If Cmd = 'R' Then
    Begin
      Write('Enter Record #: ');
      Readln(RecNum);
      FileRead ( F, Buffer, RecNum);
      For I := 1 To BUFSIZE Do
        Write(Buffer[I]);
      Writeln(Null);
    End
    Else
    If Cmd = 'W' Then
    Begin
      Write('Enter Record #: ');
      Readln(RecNum);
      FileWrite ( F, Buffer, RecNum);
    End
    Else
    If Cmd='K' Then
      FileClose ( F )
    Else
    If Cmd <> 'Q' Then
      Writeln('Unrecognized Command');
    Writeln('IoError=',IoError);
  Until Cmd = 'Q';
End.
```

Listing 6.



ResQ is quite simply the easiest-to-use, fastest and most comprehensive information management system ever developed for the IBM Personal Computer. It was designed from the ground up to provide you with the exact information you need . . . *only* the information you need . . . in the exact form you need it . . . now!

Best of all, you don't have to know a thing about computers to take full advantage of ResQ almost immediately!

Like many other Data Base Managers, ResQ organizes, files, locates, controls and interrelates just about any kind of information. It prepares reports and mailing labels, is a general filing and reference system, handles customer records and inventories, tracks sales leads and follow-ups, and does the required calculating for any of these functions automatically. But that's where the similarity to other systems stops!

ResQ is the *only* system that allows you or a secretary to design *your own* forms and reports, create them right on the screen, then have them continuously available at the touch of a key! ResQ offers password protection to keep confidential information confidential, and does it on a number of different levels. ResQ reads, searches and sorts data faster than the fastest competitor we've tested.

ResQ's larger fields hold more information, and field sizes can be easily changed to increase your number of salesmen, products, files or reports. It allows for calculations on up to 8 levels of totals at report time, and interfaces with most popular word processing programs.

The ResQ "How-To" Manual is written in plain, simple English—not like a scientific textbook in a foreign language! It was created for immediate use by clerical help who had never touched a computer before. It's been tested . . . and it works! As an added feature, "help screens" are always immediately available to answer questions for the user, without having to turn to the Manual.

We believe you'll be absolutely thrilled with ResQ and the way it prepares reports, manages information and solves your business problems. We're so certain that we offer ResQ with a 30-day money-back guarantee of satisfaction. You can try ResQ in your own business environment and prove its value for yourself . . . completely without risk!

That's a very remarkable offer . . . but ResQ is a very remarkable system!

Order ResQ through your local computer store, mail the coupon below or call . . .
(312) 298-3610

KEYSoftware Inc.™
2350 E. Devon Ave., Suite 138
Des Plaines, Illinois 60018 • (312) 298-3610

KeySoftware Inc. 2350 E. Devon Ave., Suite 138, Des Plaines, Illinois 60018

Please send_____ ResQ Software System(s) including diskettes and comprehensive, plain English Manuals at $395.00 each including postage and handling. I understand I may return these materials within 30 days of receipt for a full refund if not completely satisfied.

☐ Check or Money Order enclosed ☐ Corporate Purchase Order enclosed.

☐ Charge to my
☐ Visa ☐ MasterCard account. ☐☐☐☐-☐☐☐☐-☐☐☐☐-☐☐☐☐

Exp. Date_____ Bank No._____
(MasterCard only)

Name_____

Firm_____

Address_____

City_____ State_____ Zip_____

Signature ✓_____

Requires IBM PC, with 128 K RAM or IBM XT, 2 disk drives, DOS 1.1 or 2.0. Update policy: Return your diskette at any time and Key Software, Inc. will replace it with the latest version for a $20.00 fee.

LET YOUR PASCAL DATA TRAVEL First Class

by Brian Odlum

Many people seem to be wondering how to increase the amount of data they can address with programs written in IBM Pascal. The problem they're trying to solve arises from the fact that the architecture of the 8088 imposes a 64K size limit on any one segment, and the Pascal compiler allocates only one segment (more accurately, one group) for data. The program presented in this article should provide a few clues about how to get around this annoying problem, as well as offering a nice tool for those low-budget programmers (such as the author) who can afford only the monochrome display.

You should be advised in advance that we'll be working in Asm-86, and that you'll need an assembler to link up this program. The IBM *Macro Assembler* is the target of most of the comments in this article, but any good macro assembler for the 8086/8088 should do. The program includes documentation to allow you to follow its design. So let's get started.

The program uses portions of your computer's memory to store screen images. Four "extra pages" have been set up for the monochrome display, but with no trouble that number could be expanded to fifteen pages (per additional segment). There are two procedures ((SAVE_SCREEN and RESTORE_SCREEN) in the assembly language module that are called from a Pascal program—one to save the contents of the video buffer in memory, the other to restore the contents of that memory to the video buffer. These procedures can be very handy for such tasks as saving a screen image that you're developing in order to present a special menu, and then switching back to the original screen exactly as it was before the menu appeared.

The trick is that we have the option of declaring these "extra pages" in memory in such a way that they reside in a different segment and/or group than the data group produced by the Pascal compiler. The techniques used in this program can be used to set up as many segments as needed (limited, of course, by the amount of physical memory available and the one-megabyte limit of the machine). To keep the example simple, our program sets up only one additional segment.

How do we do it? Well, it's actually very easy; but before we get into the explanation we should probably take a quick review of segments—particularly how to declare them with the *Macro Assembler*.

The *Macro Assembler* manual refers to three attributes that can be assigned to segments; these attributes are an ***align-type***, a ***combine-type***, and a ***class***. First, let's look at the align-type.

The assembler recognizes four possibilities for the align-type: PAGE, PARA (short for paragraph), WORD, and BYTE.

BYTE means that a segment can begin at any address in memory. WORD means that a segment must begin at an even address. The distinction is important when the target processor is an 8086, which can (and usually does) fetch memory by the word. For the 8088, however, which always fetches one byte at a time, the distinction is not too significant.

The other two choices, PARA and PAGE, are probably the most commonly used. PARA directs the linker to build the segment starting on an even address, but it must be an even address whose least significant byte is 0. Thus, for example, 0EFF0H is the beginning of a paragraph, but 0EFFAH is not. If you're planning to use Debug to scan the contents of memory, it's a good idea to use the PARA attribute, because Debug dumps memory to the screen by paragraphs (if you don't ask it to do otherwise). The last attribute, PAGE, is like PARA, except that the two least significant digits of the address must both be 0. Thus, 0EF00H is the start of a page, but 0EFF0H is not.

The second category of segment attributes is the ***combine-type***.

The choices here are AT, PUBLIC, COMMON, STACK, MEMORY, or no entry.

The AT attribute is the simplest to understand. Using the AT directive, we can tell the assembler to begin a segment at a specific hardware address, so long as that address is the beginning of a paragraph. This is very useful when we want to access a specific segment in memory that has been dedicated to a special function, such as ROM or a video buffer. Our feature program declares the segment VIDEO—RAM to begin AT the address 0B000H, the start of the 4K buffer for the monochrome display.

The PUBLIC attribute directs the linker to concatenate the segment to any other segment or segments already found with the same name. Segments are concatenated in the order in which they're encountered by the linker. If a segment named DUMMY has been declared in module A and another segment named DUMMY has been declared in module B, and the second DUMMY has the PUBLIC attribute, the linker will join the two together into one segment named DUMMY and will calculate the necessary relocation values of any labels that may have been displaced by the concatenation. We're assuming that the segment in module A was encountered first by the linker. If both segments were declared PUBLIC, the concatenation would take place regardless of the order in which they were processed by the linker. (For the moment we are ignoring the *class* attribute, which, as we'll see, makes things a bit more complicated.) The final size of the concatenated segment will be the length of the individual segments added together. Note that if the segments in both module A and B are 500D bytes long, and if each contains a label at offset 100D, when the two segments are concatenated the label in the second segment will be at offset 600D. This relocation can cause problems if we have referred to that address in our code by its original numerical value, rather than by its label.

The COMMON attribute is similar to the PUBLIC attribute, with one important difference. Segments in separate modules that have the same name and are declared as COMMON are *not* concatenated but are treated as the same segment. This is handy when you want to break up a large assembly program into smaller modules and assemble them separately. The code in each module refers to the same segment, and the segment must be declared in each module in order for the module to assemble correctly, but at run time there is really only one segment being addressed. The COMMON directive tells the linker that only one segment is being allocated. If two separate segments in different modules have the same name and are declared COMMON, but are of different sizes, then the final segment after linking is the same size as the larger of the two.

The STACK directive tells the linker that the segment is part of the run-time stack segment and is therefore dedicated to a special purpose. If we are linking up an assembly language module with a Pascal driver, the compiler automatically declares a stack segment of 512 bytes for us, so we don't normally need to declare our own stack segment. If, however, we are programming completely in assembly language, we will have to declare at least one segment with the combine-type STACK, and we'll have to decide how much stack space our application will need.

The last combine-type is MEMORY. This directive tells the linker to allocate the referenced segment at a higher address than all other segments in the program. If several segments having the MEMORY combine-type are linked together, only the first one encountered is processed as a MEMORY segment; all others are processed as a COMMON segment. At least that's what the *Macro Assembler* manual says, but in practice this seems not to be true. As we shall see, the segment *class* once again makes things a bit more complicated. In any case, the last segment created by the Pascal compiler is named CONST, and so if you specify a segment with the combine-type MEMORY, it should appear after (that is, at a higher address) than the segment CONST and also higher than any other segments you may have declared. For reasons to be discussed shortly, creating a MEMORY segment is not a good idea when you're working with the Pascal compiler.

The last attribute category is *class*. The class designation is completely optional. If present, it's used by the linker to determine the order in which segments are processed. Since the Pascal compiler uses certain class names for different kinds of segments, it is customary to follow suit and use the same names in our own programs. For instance, the compiler puts all segments containing executable code into the class 'CODE'. Once the linker has encountered a segment in this class, it searches for and processes all segments in the same class before moving on to the next class. The compiler has been set up in a manner such that, ordinarily, the first segment encountered by the linker is in the class 'CODE.' That means that the linker will process all segments in the class 'CODE' before moving on to another class, and if we wish to have all the code segments in our program (that is, all segments containing executable code) to be contiguous in memory, we must put our own code segments in the class 'CODE' as well.

Likewise, the compiler uses the class name 'DATA' to designate segments that contain data. If we declare additional data segments in our assembly language module and we wish to have all the data segments in contiguous memory, we must put our data segments in the class 'DATA.' There is no reason, however, why we cannot create our own class (let's call it 'NONSENSE') and group several segments in the class contiguously in memory at link time.

At this point we encounter some of the complications alluded to earlier. It should be clear that we can create potential conflicts by assigning attributes to segments in a poorly conceived manner. Consider the following three segment declarations:

```
1) DUMMY   SEGMENT WORD  PUBLIC   'JUNK'
2) DUMMY   SEGMENT BYTE  PUBLIC   'GARBAGE'
3) SMART   SEGMENT PAGE  PUBLIC   'JUNK'
```

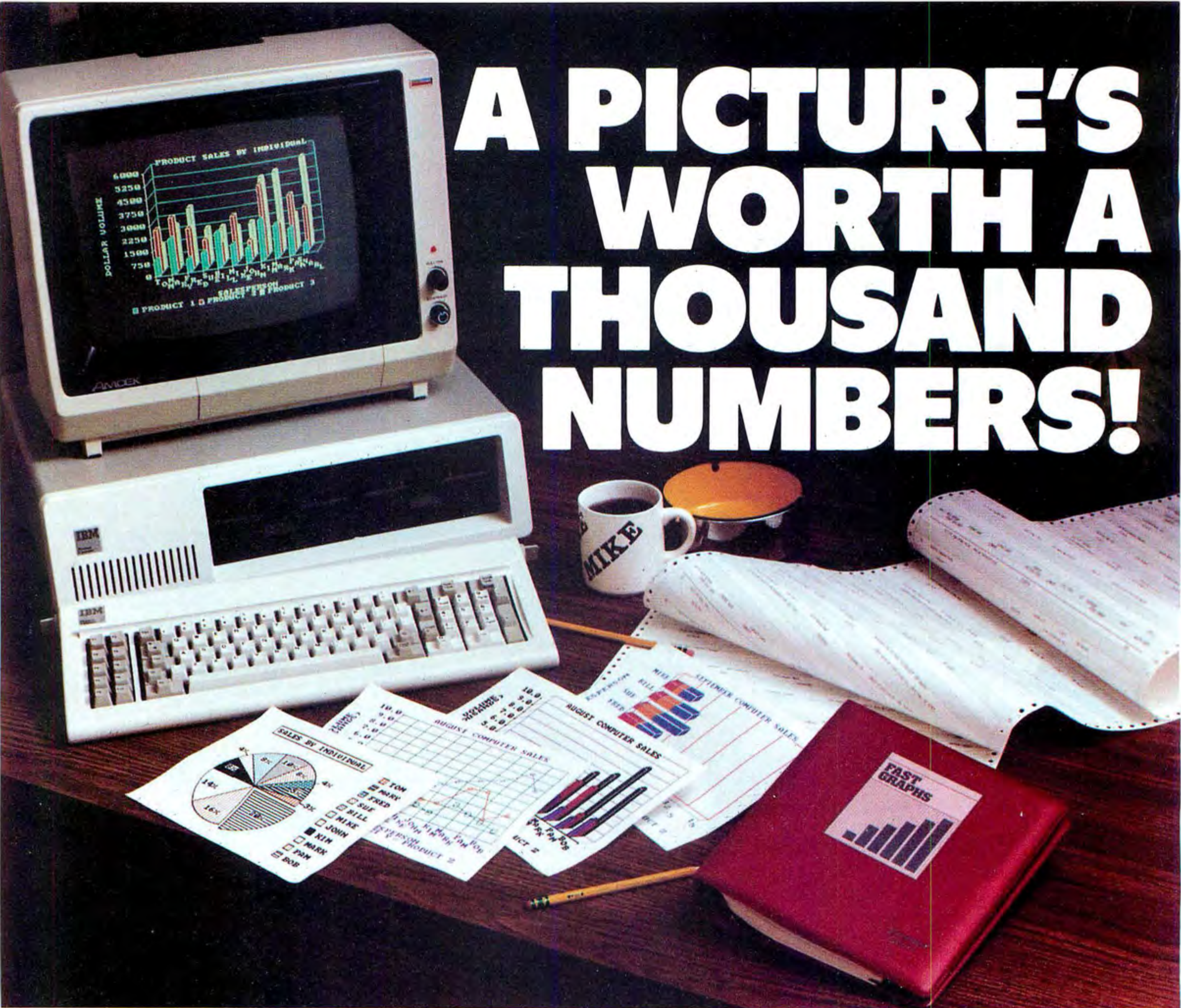
We have declared three segments, two with the same name (DUMMY), and we've assigned them all the PUBLIC attribute. The two segments with the same name are assumed to be in different modules, since the assembler will not allow a duplicate definition of a segment in the same module. Two of the segments belong in the same class ('JUNK'), but one is in a different class ('GARBAGE'). How will the linker handle these declarations?

Assuming it first encounters declaration #1, the linker will process that segment with no problem. But what next? Does it go ahead and process all segments in the same class and thus process segment #3? If it does, then it cannot concatenate segment #2 to the end of #1, as the PUBLIC attribute demands. If it does process #2 instead, then it must ignore the fact that segment SMART is declared in the same class as #1, which demands that it should be loaded contiguously in memory with #1.

In this case, the linker gives priority to the class designation, and the two segments named DUMMY will not be concatenated. There are many such conflicts we can generate by being careless or confused.

Well. So far so good. But now we encounter two subjects that make everything a bit more complicated than it really ought to be. The first complication is the subject of groups. Several of the points discussed above are not totally accurate, since up to this point we have been ignoring groups. A second complication is the subject of relocation. Few people are aware of the fact that a program generated by the Pascal compiler actually relocates a portion of itself in memory. This is not the relocation done by the program loader that is part of DOS but is instead an additional relocation done *at run time*, after control has been passed to the Pascal program.

A PICTURE'S WORTH A THOUSAND NUMBERS!



The IBM Personal Computer and Printed Reports

Everyone knows that the IBM Personal Computer does a terrific job of storing and manipulating facts and figures. Unfortunately, drawing conclusions from this information requires printed reports often running into the thousands of numbers. Now, however, you can use **FAST GRAPHS**™ to replace all those numbers with one simple to understand, pleasing to the eye, graphic report.

FAST GRAPHS™ and the "Graphic Report

FAST GRAPHS™ is the most recent addition to Innovative Software's **EXECUTIVE SERIES**.™ It converts the numbers from Visicalc® DIF™ files, manual entry, or our **EXECUTIVE SERIES**™ packages, including **T.I.M. III**® into "graphic reports". Graphic reports may be viewed on your color monitor or sent to any popular graphics printer or plotter. A graphic report can have scatter or line charts, pie charts, and two or three-dimensional bar charts, etc., all of which may be customized by the full graphic editor. This same editor may also be used to paint and draw characters or shapes to be overlayed on a graph.

Your Data Never Looked So Good

If your meetings seem to be "bored meetings" rather than Board Meetings, then **FAST GRAPHS**™ is for you. **FAST GRAPHS**™ turns dull, boring, printed reports into exciting, colorful "graphic reports". Visit your local IBM dealer today for more information about **FAST GRAPHS**™ and our other products. Or call us directly. Either way, you'll be drawing colorful conclusions in no time.

Innovative Software®

9300 W. 110th St., Suite 380
Overland Park, KS 66210 USA
913/383-1089
Telex: 209542

This relocation, as well as the group initialization of the segment registers and several other chores, is done in the module ENTXQQ, which is part of the Pascal run-time library in the file Pascal.lib. The assembly source code for this module can be found on the PAS1 disk, under the filename Entx6s.asm; it's worth studying. Briefly, all the data segments produced by the compiler are loaded in low memory by the loader in Command.com, and then they're relocated to high memory by the procedure BEGXQQ that's part of the module ENTXQQ. It is for this reason that declaring a segment with the combine-type MEMORY in one of our own modules can create problems. It can be partially or completely overlayed by the relocation process. In short, it can be a real disaster.

By making the code to ENTXQQ available to the programmer, and by placing the assembled object code in the library, Microsoft has made it possible for programmers using the Pascal compiler to modify the run-time memory image of a Pascal program. The task is not easy, however, and it is made even more difficult by a notable absence of documentation. What little there is can be found in Appendix D in the manual, but it is hopelessly inadequate. We are left to the slow and uncertain process of trial and error. To delve into this modification in detail is beyond the scope of this article, but as Pascal programmers we should be aware of all the cards on the table. For those inquisitive souls out there who like challenges, a very brief outline is in order.

Make the changes you want to the ENTXQQ module. You must include the procedure DOSXQQ, even if it is not called from your own code (the compiler generates code that apparently makes internal calls to this routine, so it must be there). Assemble the modified module, and at link time explicitly declare it as one of the object modules you wish to include in the linker session. Since the linker will only search the library for unresolved externals when it cannot find them elsewhere, the ENTXQQ module in the library will be ignored and your modified module will be substituted. Take care that all labels, names, and so on that were exported from the original ENTXQQ module by the PUBLIC pseudo-op are also exported from your modified module. Otherwise the linker will find one or more unresolved externals somewhere and will load both your version *and* the original version.

For the rest of us who may not be so courageous, take heart. We can still set up additional data segments without going into the labyrinth just described. We must still understand a little about groups, however—a subject that will be easier to grasp if we first take a closer look at our feature program so we can have a concrete example to help us make sense out of all the theory.

The assembly language module declares two additional segments, one named VIDEO_RAM and the other named PAGES. As mentioned earlier, VIDEO_RAM uses the AT directive. The align-type is not used in this declaration, since PARA is implied; in actuality the segment begins on a page boundary. Nor is there a class entry, since the segment is bound to a hardware address and is therefore not relocatable. The linker only uses the class name to group segments contiguously in memory, which it cannot do if the segment cannot be moved.

The other segment, PAGES, has two attributes declared. Its align-type is PAGE, and its class is 'SCREEN'. That means the segment will start on an even address whose two least significant digits are 0. The program creates a new class just to demonstrate that it can be done.

Now let's put together the .exe file. Compile the source code for program PAGE—DRIVER and assemble the source code for module PAGE—CODE. You should now have two .obj files, with names you have given them, which should both be on the same disk. Put that disk in drive B, set the default drive to B, put the disk with Link.exe and Pascal.lib in drive A, and type A:LINK < assembly code filename > + < Pascal code filename >. Notice that the module containing the segment PAGES was named first. This is critical. When the linker prompts you for the name of the list file, type CON (for console). After a moment, a listing of all the segments in the complete program will be written to the screen, with their hexadecimal starting and stopping offsets and their sizes. It should look something like this:

Start	Stop	Length	Name	Class
00000H	04007H	4008H	PAGES	SCREEN
04010H	04064H	0055H	PAGE_CODE	CODE
04066H	0417FH	011AH	PAGE_DRIVER	CODE
05E70H	05F36H	00C7H	ENTXQQ	CODE
08790H	08790H	0000H	HEAP	MEMORY
08790H	08790H	0000H	MEMORY	MEMORY
08790H	0898FH	0200H	STACK	STACK
08990H	09063H	06D4H	DATA	DATA
09070H	097ADH	073EH	CONST	CONST

Origin	Group
F97B:0	DGROUP

The code segments generated by the compiler to do initialization, run-time error checking, and other related chores have been removed from this list. Note that the compiler uses the same word—*data*—to designate a segment name and a segment class. The compiler also uses the word *memory* as a segment name and a class name, neither of which should be mistaken for the assembler's combine-type MEMORY. This can be confusing. Undoubtedly there's a good reason why the folks at Microsoft chose those names; but it's not exactly clear what the reason was.

Okay, let's talk about groups. A group is a collection of segments that, when placed together, form a contiguous block of memory no larger than 64K. Individual segments are named to a

```
         PAGE      64,132              ;define listing format
         TITLE     SCREEN_PAGES        ;names the module as well
         .XALL                         ;limited macro expansions

;------------Macro definitions

SET_UP   MACRO     XXX,YYY
         ASSUME    DS:XXX,ES:YYY       ;so assembler can check addressability
         PUSH      BP                  ;save caller's frame pointer
         MOV       BP,SP               ;set up callee's frame pointer
         PUSH      DS                  ;save caller's DS
         PUSH      ES                  ;save caller's ES
         CLI                           ;disable interrupts to load seg registers
         MOV       AX,XXX              ;get new segment value
         MOV       DS,AX               ;load into DS
         MOV       AX,YYY              ;get new segment value
         MOV       ES,AX               ;load into ES
         STI                           ;enable interrupts
         ENDM

TRANSFER MACRO     XXX,YYY,ZZZ
         MOV       XXX,ZZZ[BX+SI]      ;beginning offset of selected page
         MOV       YYY,OFFSET CRT      ;beginning offset monochrome buffer
         MOV       CX,VIDEO_BLOCK/2    ;number of words to transfer
         CLD                           ;clear direction flag, forward
         REP MOVSW                     ;make the transfer
         ENDM

ALL_DONE MACRO     XXX
         POP       ES                  ;restore caller's ES
```

Listing 1 continues on following page.

group by the GROUP pseudo-op, and the group is given a unique name. There are five data segments created by the Pascal compiler that are named to a group (named DGROUP). They are HEAP, MEMORY, STACK DATA, and CONST. These five segments, as we can see from the memory map produced by the linker, are all contiguous in memory. Notice also that HEAP and MEMORY appear together, because they belong to the same class 'MEMORY' (don't get confused here!). One reason for naming segments to a group is the ease with which different groups can be set up to be addressed by the executable code. With the Pascal compiler, the stack segment was included in the group, and thus the SS, DS, and ES registers are all initialized to the same base address of a 64K block of memory. This simplifies certain kinds of operations. For instance, there is no need to pass the segment base of an argument in a procedure or function parameter list (as is always done with IBM Fortran), since all variables, even if they exist within different segments, have the same segment base value—the value of the group. In IBM Pascal, if we want to pass both the segment base and the offset of an argument, we must explicitly use the VARS declaration. The VAR declaration will default to passing the offset only, because the segment base is implied by the grouping of the data segments.

Grouping segments can produce some unexpected results. Notice, for instance, that if the stack and data segments have been named and initialized to a group, and if the BX and BP registers both contain the same value, the instructions MOV AX,[BX] and MOV AX,[BP] are identical. That is, they will both dereference the same physical address. It may be that for a recursive, stack-oriented compiler, such as Pascal, this is a necessary condition for the compiler to do its job correctly when the target processor is a "segmented" architecture such as the 8086/8088. If the stack segment and data segment had *not* been named to a group, the two instructions would not necessarily be identical, since the BX register defaults to the segment pointed to by the DS register and the BP register defaults to the segment pointed to by the SS register. Also, if the DS and ES registers contain the same value, as a result of grouped segments, the assembler instructions to manipulate strings (such as MOVS and CMPS) may not work exactly as intended. That's because these instructions use the SI register to point to the source segment (DS) and the DI register to point to the destination segment (ES), which are now the same physical block of memory. We just have to look before we leap.

So what does all this really mean? Well, for one thing, it means that if we try to extend Pascal's DATA segment by declaring a segment named "DATA" that is PUBLIC and in the class 'DATA', and in the process we push the size of the group over 64K, our program will not link up correctly. The linker will give us a message like "fixup offset exceeds field width," telling us that we have pushed something too far away in memory and that it can no longer be reached by the code. So if we want to declare something of significant size, such as "extra pages" for saving screen images, to be safe we must declare the new segment outside Pascal's DGROUP. But wait a minute. Notice that the very first segment listed in the memory map, on the other side of the code segments, is the segment PAGES in the class 'SCREEN'. We've already done it. But how?

The explanation is simple. The first module that the linker encountered was our assembly language module, and the first segment it encountered within that module was the segment PAGES. So it began with that segment instead of the 'CODE' segments. Had there been other segments named to the class 'SCREEN' in this or other modules, they would all have been loaded contiguously in memory before any of the 'CODE' segments.

Well, we now have a new segment outside of DGROUP, but we also have a new problem. At run time, the SS, DS, and ES registers point (usually) to the DGROUP produced by the compiler. Our two assembly language procedures do business, however, with the segments declared in our module, which means the DS and ES registers must be changed. We must therefore set up the addressing to

```
                POP         DS                  ;restore caller's DS
                POP         BP                  ;restore caller's frame pointer
                RET         XXX                 ;discard parameters and return
                ENDM

;--------------Stack frame for access to parameter

FRAME           STRUC
SAVE_BP         DW          ?                   ;caller's frame pointer
RETURN_ADDRESS  DD          ?                   ;pushed by Pascal
PAGE_NUMBER     DW          ?                   ;user's page selection
FRAME           ENDS

;--------------Equates and Publics

VIDEO_BLOCK     EQU         1000H               ;size of video buffer (bytes)
                PUBLIC      SAVE_SCREEN
                PUBLIC      RESTORE_SCREEN
                PAGE

;------------- Segment to access monochrome CRT

VIDEO_RAM       SEGMENT     AT 0B000H
CRT             DB          VIDEO_BLOCK DUP(?)
VIDEO_RAM       ENDS

;--------------Segment for extra pages

PAGES           SEGMENT     PAGE 'SCREEN'
PAGE1           DB          VIDEO_BLOCK DUP(?)
PAGE2           DB          VIDEO_BLOCK DUP(?)
```

Listing 1 continues on following page.

these segments within the module and restore everything back to the original values before returning to the Pascal program. The two macros SET_UP and ALL_DONE serve this purpose nicely. If you do a lot of Asm-86 programming in conjunction with Pascal, you will probably find these two macros to be of great practical value. Note that interrupts are disabled when the segment registers are loaded; this is to avoid a potential system crash. The third macro, TRANSFER, was included only as an example of the use of null labels in a macro expansion.

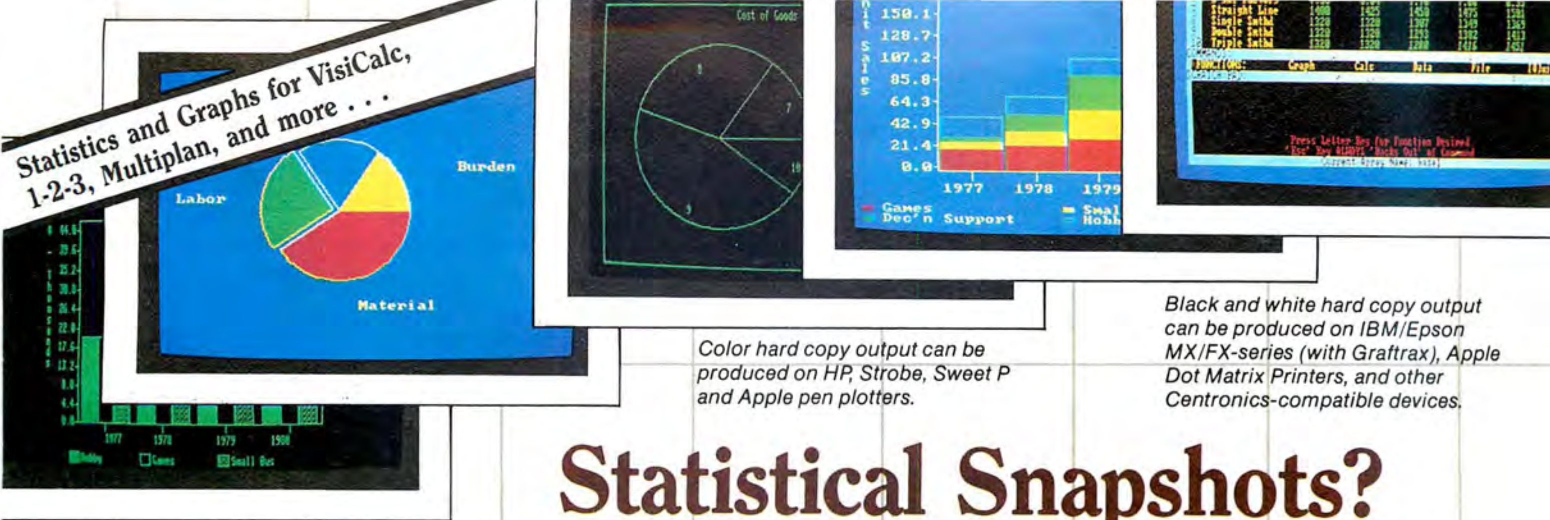
Notice that in procedure SELECT_PAGE, it appears as though we could have written MOV BX,4000H, instead of MOV BX,OFFSET PAGE_INDEX. After all, 4000H is the offset of the label PAGE_INDEX, and a lot of mathematical ability is not required to figure that out. But notice that the offset is only static so long as the segment PAGES is not combined with any other segment by the linker (as might have been the case if we had declared it PUBLIC and another segment in some other module had the same name and class). Using the OFFSET operator and the label name forces the assembler to flag the actual value assembler as relocatable; at link time the appropriate value is calculated and patched into the object code. This makes our code easier to change and less prone to error.

The Pascal driver included with the program is pretty unimaginative but suffices as an example of how to declare and call the interesting stuff.

One final note: There are two annoying bugs in my assembler, which may also exist in yours. The first is that if line 1 of the source file is not blank, the listing looks like a patchwork quilt on the first page. The other has to do with macro expansions. The assembler directive .LALL, which is supposed to tell the assembler to expand everything in the macro (including lines that don't produce object code), also seems to toggle double line spacing in the listing. Hence our program has the directive .XALL instead, and the ASSUME pseudo-op in macro SET_UP does not appear in the macro expansion. ▲

```
PAGE3          DB          VIDEO_BLOCK DUP(?)
PAGE4          DB          VIDEO_BLOCK DUP(?)
PAGE_INDEX     DW          PAGE1
               DW          PAGE2
               DW          PAGE3
               DW          PAGE4
PAGES          ENDS
               PAGE

;-------------Code segment

PAGE_CODE      SEGMENT     PARA 'CODE'
               ASSUME      CS:PAGE_CODE

;-------------Procedure SELECT_PAGE

COMMENT * This procedure sets up the BX and SI registers with the necessary values to access
the correct starting offset of the selected page.        *

SELECT_PAGE    PROC        NEAR                     ;local procedure
               MOV         SI,[BP].PAGE_NUMBER      ;get user's argument for page number
               DEC         SI                       ;because the base is 0, not 1
               SAL         SI,1                     ; * 2 to account for word size (2 bytes)
               MOV         BX,OFFSET PAGE_INDEX     ;get offset of translation table
               RET                                  ;return to caller
SELECT_PAGE    ENDP                                 ;end of procedure SELECT_PAGE

;-------------Procedure SAVE_SCREEN

COMMENT * This procedure saves the contents of the monochrome CRT in memory. Pascal's
declaration is :

    PROCEDURE SAVE_SCREEN(PAGE_NUMBER:INTEGER);EXTERN; *

SAVE_SCREEN    PROC        FAR                      ;far call from Pascal
               SET_UP      VIDEO_RAM,PAGES          ;macro
               CALL        SELECT_PAGE              ;set up addressing
               TRANSFER    DI,SI,ES:                ;macro
               ALL_DONE    2                        ;macro
```

Listing 1 continues on page 114.



Color hard copy output can be produced on HP, Strobe, Sweet P and Apple pen plotters.

Black and white hard copy output can be produced on IBM/Epson MX/FX-series (with Graftrax), Apple Dot Matrix Printers, and other Centronics-compatible devices.

Graphics output includes line charts, side-by-side and stacked bar charts, and pie charts. IBM PC version includes a high/low/close/volume securities chart. The Apple version includes a combined bar/line chart function and has a semi-log scaling option.

Statistical Snapshots?

Accelerate your trend forecasting in electronic spreadsheets with a revolutionarily simple visual statistics system: Graph 'n' Calc

The Missing Link. Until now, VisiCalc, Desktop Plan, Multiplan, and other DIF-accessed electronic spreadsheets have lacked two crucial functions: forecasting statistics and user-fluid 'analytical' graphics.

In other words, meaningful pictures that are derived from fact and time-proven analytical methods.

Now, with Graph 'n' Calc, you can transform wild-guess *"what if"* scenarios into statistically derived *"what's probable"* models. That's a breakthrough, whether you presently have a spreadsheet program or not.

Never has statistical power been so fluid, easy to apply and intelligently focused.

"A TRULY ELEGANT SYSTEM". Independent reviewers rave about Graph 'n' Calc, its balance of power and ease of use.

By taking complex formulae and reducing them to a single keystroke, now even 'hunt and peck' business executives can perform: multiple linear regressions, exponential smoothing, time series and seasonality factoring, advanced mathematical functions, and even user-supplied BASIC subroutines, all conveniently tucked away in an easy access menu.

Specifically designed to share data files with DIF-accessed spreadsheets such as VisiCalc and 1-2-3, Graph 'n' Calc lets you extract rows and columns of numbers, statistically manipulate them, improve their accuracy, and pass them back to your other spreadsheets.

Goal-Seeking Graphics. Unlike any other spreadsheet or graphics packages today—integrated or stand alone—Graph 'n' Calc *lets you think in a visual domain!* How?

By streamlining the drawing routines, adding 'best guess' statistical defaults and a quick escape command structure. This means that graphs are easy to draw, redraw and draw again . . . in seconds.

Human beings naturally process pictures thousands of times faster than words and numbers.

With Graph 'n' Calc, you now have the ability to visually process normally tough numerical values such as trends, seasonal fluctuations, and rates of growth. Independent research suggests this visual approach can increase decision analysis productivity 20 fold!

"EERILY ACCURATE". That's what experienced VisiCalc and 1-2-3 users have said about Graph 'n' Calc.

"Graph 'n' Calc doesn't replace your electronic spreadsheets, it just makes them more accurate and saves hours and hours of work building probability forecasts," says another software magazine.

Come see for yourself, the "Polaroid" equivalent of the statistical graphics systems at your nearest computer store.

Color and monochromatic video display versions are available for the IBM PC and Apple systems, respectively.

DESKTOP COMPUTER SOFTWARE INC.

Graph 'n' Calc

303 Potrero St. 29/305
Santa Cruz, CA 95060
Telephone (408) 458-9095

Name ______________________

Firm/Title ______________________

Address ______________________

City/State/Zip ______________________

Phone ______________________

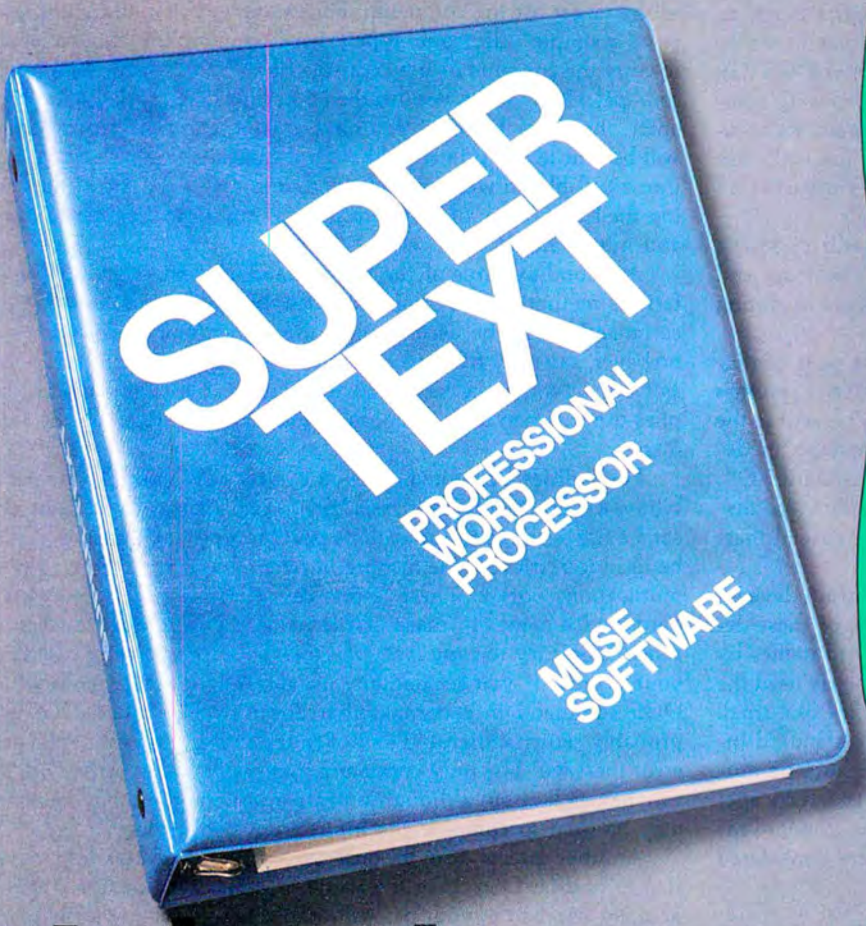
```
SAVE_SCREEN        ENDP                                    ;end of procedure SAVE_SCREEN
                   PAGE

;-------------Procedure RESTORE_SCREEN

COMMENT * This procedure restores the contents of a page saved in memory to the
monochrome display. Pascal's declaration is:

  PROCEDURE RESTORE_SCREEN(PAGE_NUMBER:INTEGER);EXTERN; *

RESTORE_SCREEN     PROC        FAR                         ;far call from Pascal
                   SET_UP      PAGES,VIDEO_RAM             ;macro
                   CALL        SELECT_PAGE                 ;set up addressing
                   TRANSFER    SI,DI                       ;macro
                   ALL_DONE    2                           ;macro
RESTORE_SCREEN     ENDP                                    ;end of procedure RESTORE_SCREEN
PAGE_CODE          ENDS                                    ;end of code segment
                   END                                     ;end of assembly
```

Listing 1.

```
($LINESIZE:132)
($PAGESIZE:64)
($DEBUG-)
($SYMTAB-)
($TITLE:'PAGE_DRIVER')

PROGRAM PAGE_DRIVER(INPUT,OUTPUT);

CONST
        ROWS_PER_PAGE              = 24;
        COLUMNS_PER_PAGE           = 80;
        NUMBER_OF_PAGES            = 4;

TYPE
        PAGE_RANGE                 = 1..NUMBER_OF_PAGES;

VAR
        I,J,K     :INTEGER;
        DONE      :BOOLEAN;
PROCEDURE SAVE_SCREEN(PAGE_NUMBER:PAGE_RANGE);EXTERNAL;

PROCEDURE RESTORE_SCREEN(PAGE_NUMBER:PAGE_RANGE);EXTERNAL;

BEGIN
FOR I := 1 TO NUMBER_OF_PAGES DO
        BEGIN
        FOR J := 1 TO ROWS_PER_PAGE DO
                FOR K := 1 TO COLUMNS_PER_PAGE DO
                        WRITE(CHR(I+ORD('0')));
        SAVE_SCREEN(I)
        END;
DONE := FALSE;
REPEAT
        BEGIN
        WRITELN('Input a page number between 1 and ',NUMBER_OF_PAGES:1);
        READLN(I);
        IF (I IN [1..NUMBER_OF_PAGES]) THEN
                RESTORE_SCREEN(I)
        ELSE
                DONE := TRUE
        END
UNTIL (DONE);
WRITELN('Normal Termination');
END.
```

Listing 2.

SUPER-TEXT Professional does everything the competition does, except one thing.



ON-LINE "HELP"

ON-SCREEN FORMATTING

It doesn't cost nearly as much.

SUPER-TEXT PROFESSIONAL FOR IBM* provides the most powerful word processing features on the market today at a sensible price. Super-Text makes word processing easy. A few of its many features are valuable **On-Screen Formatting**, easy to use **On-Screen Help Reference Guides**, professional **Split-Screen capability**, and **Autolink.**

Super-Text Professional utilizes all of the IBM function keys. It provides automatic page headers and footers, automatic page and chapter numbering, and the Preview Mode allows you to check page endings.

Super-Text provides for easy text editing. Special block operations for text copy, save and delete. Automatic tabbing and formatting plus multi-file search and replace. User definable characters allow you to make full use of most printer functions, including multi-color printing, superscripting and subscripting. Plus Super-Text provides full compatibility with IBM Edlin Files.

This very powerful and easy to use word processor can be yours for only **$175.00.**

*Super-Text requires IBM DOS Version 1.1., minimum 64K. Also available for Apple II+ and Apple IIe.

INTRODUCTORY OFFER! $99.

MUSE® SOFTWARE

347 North Charles Street, Baltimore, Maryland 21201 301/659-7212

SYSTEM NOTEBOOK

by Alan Boyd

Last month's column promised a look at the concept of *pipes*, an especially attractive feature of DOS 2.0. Before we get to pipes, however, we need to understand a more basic feature of DOS 2.0: the facility to redirect input and output. I/O redirection, as it's commonly called, is simply the ability to substitute transparently one input device for another, via a DOS command. Since pipes make use of I/O redirection, an understanding of this simpler feature will lead directly to an understanding of pipes.

As with many of the facilities of DOS 2.0, I/O redirection is a feature found in many mini and mainframe computer systems, and is, in particular, a hallmark of Bell Labs's popular Unix operating system.

"Standard I/O." In the normal or default state, the pc is configured as a "console"; that is, the operating system always assumes that the input comes from the keyboard and the output is sent to the video screen. Because these two devices are the standard, or default, input and output devices, they are also called "standard I/O": The keyboard is referred to as "standard input," and the video display as "standard output." We'll talk later about a very special characteristic of standard input and standard output.

There are, of course, numerous other input and output devices, such as disk drives and printers. Under DOS, several of these devices receive special treatment and are given predefined names by which they may be accessed. In the past, we have regularly used the command *copy con: file* for creating "on the fly" files like small batch files. Con: is the special name assigned to the standard input/output devices; on input, Con: refers to the keyboard; on output, Con: refers to the video display. Con: can be considered a special name for a special case, the special case being the standard input and output devices. Some other devices that are considered special:

Device	Name	Stands for
Keyboard	CON:	Console
Screen	CON:	Console
Asynchronous Port #1	COM1: or	Communications port #1
	AUX:	Auxiliary device
Asynchronous Port #2	COM2:	Communications port #2
Parallel Port #1	LPT1: or	Line printer #1
	PRN	Printer
Parallel Port #2	LPT#2:	Line printer #2
Parallel Port #3	LPT#3:	Line printer #3
Dummy device	NUL:	Null device

The names associated with each of these devices are reserved words under DOS; that means you cannot create a file name with the same name as any of these device names. In return for that inconvenience, DOS lets you apply any operating system command to them in exactly the same way you would apply it to a file name. The devices that have been given special names are called *standard devices*. Con: is thus both a standard device and the standard I/O.

The *copy con: filename* example that we have been using tells DOS to treat the Con: device—meaning, in this case, the keyboard—as the input file for the command. We could just as easily have designated another standard device, as opposed to a disk file, as the target for the command. For example, you could set up your pc as a simple "electronic typewriter" by using the *copy* command to copy the contents of the "console file" to the "printer file": *copy con: lpt1:*. Now when you hit the F6 key (or hit control-Z) and press enter, the characters that you have entered from the "console file" will be sent to the printer. Although this facility doesn't turn your pc into a full-blown word processor, it does do a good job of illustrating the functioning of standard devices and how they may be used and interchanged.

A second example of the use of standard devices gives us an alternate method of viewing disk files. Normally, the *type filename* command would be used to view the contents of a file; we could, however, say *copy filename con:* if we wished to display the file. Just as Con: refers to the keyboard when used in a command that implies input, when used in an output command it refers to the video display.

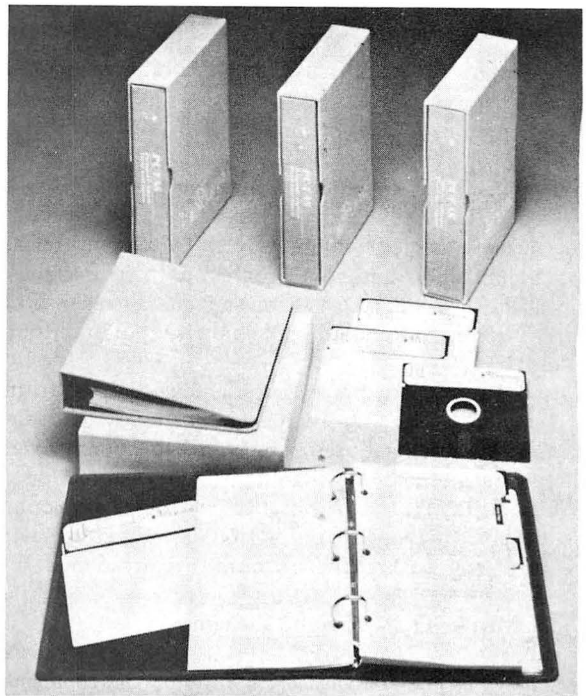
Possibilities Unlimited. As you can by now imagine, the possibilities are practically limitless. *Copy filename com1:* could be used to send a file to the communications port, or *copy com2: filename* could be used to retrieve a file coming in over a modem attached to communications port 2. There is one caveat, however, to using standard device names in place of file names: The context must make sense. Don't try to *copy lpt1: filename*, because you can't read from your printer. If you accidentally use a standard device name with a DOS command in a context that doesn't make sense, DOS will probably return either a "File not found" or an "Illegal parameter" error message. The only command that really makes sense with device names is *copy*, although on rare occasions the *comp* command may be used to compare the input from two devices.

Another precaution to observe when using standard device names is this: Don't use the name of a standard device that isn't actually attached to your pc. As the IBM manual rather tartly points out, the results can be "unpredictable." There is one exception to this rule, and that concerns the *nul:* device: This standard device is the only one that does not equate to a real-life piece of hardware. *Nul:* is a convenience for programmers to use when testing and debugging. For example, testing a piece of code that drives an output device such as a printer can be very slow and tedious; by substituting the *nul:* standard device for the real device, the programmer can greatly speed up the debugging task. That's because when the *nul:* device is used for input, it simply generates an EOF (end-of-file) marker as soon as it is called, thereby short-circuiting the input stage. When called as an output device it simulates all the actions involved in receiving data, although no data is actually sent.

The *nul:* device can also help determine whether a problem is being generated by hardware or software. Replace suspect with *nul:*; if the software runs without a hitch, it's likely the problem was in the device. If the result is the same with *nul:*, it's likely that the problem is with the software.

The capability of redirecting I/O from one standard device to another is a very useful feature of DOS 1.0 and 1.1, but it's just a prelude to the powers of DOS 2.0. DOS 2.0 allows dynamic redirection, which means that programs that conform to one simple rule can have their input or output streams temporarily redirected to just about anywhere that makes sense. And all you need to do to redi-

You Deserve The BEST



FIXED ASSET SYSTEM

PC/FAS is an easy-to-use record-keeping system that makes it simple for you to have complete and accurate records of your property and equipment for the IRS, state and local governments, insurance companies, and your business auditors.

PC/FAS automatically calculates ACRS and five other types of depreciation and investment tax credits needed for you to get the full value from new, lucrative depreciation tax laws.

Its wide selection of reports gives you complete management information on up to 99,999 owned or leased items of property or equipment in your office, plant, home or fleet. **PC/FAS** reports on each item by location, custodian, date, vendor, manufacturer, etc.

PERSONAL FINANCE PROGRAM

PC/PFP is the **Best** cash management system for your business or home for only $95.00. This best-selling program is exciting because it shows you through a large choice of pre-formatted reports, bar graphs, and on-the-screen displays where your money comes from and where it goes. Businesses have found the precise, easily accessible, and complete summaries and reports of all receipts and income are invaluable management tools that can easily **(and less expensively)** be transformed by CPA's into tax reports, income statements, and balance sheets.

HERE'S WHAT OUR REVIEWERS SAY:

"Best Program's PC/PFP is first class all the way. Any company looking to sell to the personal computer market should take Best's whole package as a model—this is the way it should be done. Best Programs offers a superior package in all respects. Best is best of this lot."
Corey Sandler PC Magazine, Vol. 1, No. 10

"Well-documented . . . program runs smoothly, and always presents clear, easy-to-understand prompts. Functionally, PC/PFP is complete. The program is well done and a good value."
Will Fastie Creative Computing, January 1983

"My overall impression of PC/PFP is a good one . . . suits the needs of a great number of people who want simple, straightforward help in their personal finances."
Cliff Smith, C.P.A. Capital PC Users Group Monitor

WE LOVE TO READ OUR MAIL . . . YOU WOULD TOO!

"Brilliant software! Why? Because its simple. Covers all needed uses. You deserve much credit as everything is so well planned out. Price is most attractive."
Mr. Erwin Jackson, Jr. Charlotte, North Carolina

"I consider PFP to be one of the finest programs for the IBM PC."
Barry N. Parsley, M.D. Lexington, Kentucky

"I have bought programs since 1977 and I can unequivocably say that PC/PFP is the best buy for the money I have ever made. Please keep developing programs. The industry needs your high standards, and your reasonable prices."
Mr. James Gore Creskill, NJ

"The program works well and is easy to use. The newsletter is good and I must say the level of customer support couldn't be better."
Lorne Sheren, M.D. Staten Island, NY

"Your User Report is hot . . . support from software publishers is a welcome help."
Michael Truffer Deltona, Florida

The Quality Software Company

BEST PROGRAMS

5134 Leesburg Pike, Alexandria, VA 22302 (703) 931-1300

rect I/O is type a few extra characters on an otherwise conventional command line.

"Aha!" you say. "One simple rule—I've heard that before." One can't blame you for being suspicious. But in this case, the simple rule is really simple: Any I/O in your program that takes its input from standard input (that's the keyboard) or sends output to standard output (the display) will be affected by dynamic relocation. If you want all of the I/O in your program to be redirectable, simply write it all to standard I/O. Wherever it makes sense, DOS commands are written to standard I/O, so their input or output is redirectable.

Output Redirection. Since output redirection is a little more obvious than input, let's start with some examples that redirect output. One of the principal uses of output redirection is to send output that would normally have gone to the screen to a disk file instead. This facility can be very useful in software development and debugging, since it allows all output to be captured for review. The general format of a command for output redirection is:

A>COMMAND > d:path \ filename

where the A > is the standard DOS prompt and *command* represents a DOS command or a program. The second > is the redirection symbol (unfortunately the same as the symbol used in the standard DOS prompt); it tells DOS to take all output that would have gone to the screen ((the standard output) and instead send it to the file or device described by the following parameter. (The example just shown has the output redirected to a file, but it can just as well be redirected to any standard device that can accept output.) You can think of the output redirection symbol as an arrow that points to the new target for the output.

A simple but very useful example of output redirection is a directory command that sends its output to the printer rather than the screen:

A>DIR > LPT1:

Likewise, the output target could be a filename; any valid filename can be used. As another example, a report generated by a spreadsheet could be sent to a disk file to be formatted and dressed up later with a word processor, or for inclusion in a longer report.

I/O redirection is especially useful in systems work; it's one of the many features that have endeared Unix to programmers everywhere. If, for example, there was some trouble with a program and the computer was at a location other than that of the original programmer (the usual situation, actually), the output of the program could be redirected to a modem port, and the programmer at the other end of the line could diagnose the problem remotely. (Or that's the theory, at any rate.)

Occasionally you'll run into a problem trying to use I/O redirection with a commercially available program. That's because the programmer bypassed the I/O mechanisms of the operating system and talked directly with hardware. Such programs make it impossible to use DOS facilities and are a further argument for programmers to quit making end-runs around DOS. In extreme cases, such programming practices make it impossible to use an existing program at all with new versions of DOS.

When an output redirection command is entered, DOS first checks to see if the designated file exists; if it doesn't, DOS will create the requested file. If the file already exists, DOS will overwrite it by setting the write pointer to the beginning of the file before sending data to the file. That means that, as with any process that writes to a file, you should be careful how you use output redirection.

It also means there is a need for a version of output redirection that appends new data to the end of the file, rather than overwriting the file. DOS has such a feature: Simply use two > symbols instead of one. For example, if you want to add a directory listing to an existing file, use the command

A>DIR > > FILENAME

If the file exists, the directory listing will be appended to its end; if it doesn't exist, it will be created and the directory written to it. It does not make sense to try to append information to a standard device.

As we noted previously, output redirection will work with any program that runs under DOS and makes use of standard I/O. As another example, let's apply it to *Mortgage*, one of the sample Basic programs on the DOS distribution disk. First, format a blank disk and copy the files Basic.com and Mortgage.bas onto it. Then log onto the new disk and enter the following command:

B>BASIC MORTGAGE > MORT.TXT

This command tells DOS to redirect the output of the Basic program *Mortgage* from the screen to a file called Mort.txt. All output that would have gone to the screen is now available for review in Mort.txt.

How the Other Half Lives—Input Redirection. The other half of the I/O redirection story is input redirection, a very powerful feature that can be a little trickier to use than output redirection. Input redirection allows you to specify input to come from a source other than the keyboard, the standard input. Input redirection can be particularly useful when a second terminal is attached to the pc via a serial port: The source of input could be easily redefined as the standard device (for example, Com1:) to which the terminal was attached. To redirect input to Com1:, give the command

A>COMMAND < COM1:

The < symbol indicates input redirection; you can remember it by thinking of it as an arrow pointing to where the input will go.

Of course, it's also possible to have input come from a disk file. Let's use the *format* command as an example. In its simplest form, the format command

A>FORMAT B:

will format the disk in drive B:. DOS issues a prompt:

Insert new diskette for drive B:
and strike any key when ready

DOS then pauses and waits for the user to press a key before formatting the disk. When it has finished formatting, it prints its standard report and then issues the prompt

Format another (Y/N)?

It then waits for either the Y or N key to be pressed.

Now, if we closely examine the interaction between the user and the computer, we see that the first prompt was answered by pressing any key—let's assume enter was pressed—and the second prompt was answered with either a Y or N. That means that if we had wanted to format the disk twice (dumb, but don't worry about it) and then exit the *format* command and return to the operating system, we would press in sequence: Enter—Y—Enter—N. If we created a file that contained these keystrokes and redirected input to that file, we would essentially have created a process that automatically would've formatted a disk twice. Try it by

A>COPY CON: TEST

Then type the characters enter, Y, enter, N, and the F6 (or control-Z) key and enter to close the file. Now run *format* with redirected input:

```
A>FORMAT B:>TEST
```

Without stopping, DOS forges ahead and formats the disk twice, with no waiting for keyboard input. When combined with other DOS capabilities such as batch files with replaceable parameters, this can be a very powerful feature that can be used to define some very complex sequences of events.

As a final example, let's use redirection in conjunction with a simple little program written in Basic. (If you hate Basic, have patience; the program is by way of example only. When you're done with this column, you should be able to use redirection wherever you want.) The Basic program collects four names entered from the keyboard and stores them in a small array. Here's the program:

```
10  CLS: DIM N$(4)
20  FOR I = 1 TO 4
30  INPUT N$(I)
40  NEXT I: PRINT : PRINT
50  PRINT "The four names are:" : PRINT
60  FOR I = 1 TO 4
70  PRINT N$(I)
80  NEXT I
90  END
```

Check the program by running it; then, since we will be using it later as part of a more complex example, save it under the name "Names."

Operator–Program Interaction. Now, let's take a close look at the interaction between the operator and program. The input to the program consists of four names, each terminated by a carriage return (enter). If we created a file containing four names and defined it as the input for the program, the whole process could be automated. Use

```
A>COPY CON: LIST
```

to create a file called List, enter four names terminated by a carriage return, and close the file with the F6/enter combination. Now we can execute the command

```
A>BASIC NAMES  < LIST
```

and look at the results. The program did not halt for input, reading instead the contents of the file list. An interesting side effect of having input redirected to a file is that, upon completion, control is returned directly to the operating system.

Let's tie it all together by using this little program with both input and output redirected. The input will be from the file List that we created earlier; the output will be directed to a file called Report. Enter the command

```
A>BASIC NAMES  < LIST  >REPORT
```

When you examine the file Report, you will see that it is a complete record of everything that happened on the screen. This demonstrates an excellent method of tracking down problems in a program by capturing the interaction for future examination. It also automates and speeds up program testing.

Redirection can be used to simulate the piping capabilities of Xenix or Unix. Based on what we now know about redirection, we're ready to tackle both pipes and filters, which we will do in next month's column. ▲

If you think you need to spend $700 for a good data base management system, don't let our $225 price tag stop you.

Surely we can put the extra $475 to good use.

Undoubtedly you can, too.

Data Design™ is a full-feature, relational data base management system (DBMS) for people who value their time as well as their money. Written expressly for the IBM Personal Computer, Data Design takes advantage of the special function and editing keys so you spend your time working, not learning.

Waiting for a sort could put you out of sorts.

Data Design's unique internal structure means you retrieve data instantly without ever waiting for a sort. Print an alphabetical address list and then zip-sorted mailing labels with no delay.

Data Design even lets you enter and print information on forms you design to look just like the ones you use everyday—with all calculations done automatically.

While other DBMS users are getting sorted out, you can be getting results in just the formats you need.

It helps to memorize one key word:

Press F1 whenever you need help and you'll be greeted by a tutorial that addresses whatever you're working on *at that moment!* While users of other popular DBMSs are still memorizing program commands, you'll be sped along your way by menus and comprehensive help screens.

In one disk and out the other.

Data Design is the only major DBMS that allows telephone transfer between any two Personal Computers, so you can communicate instantly and accurately across the street or across the country.

Data Design also lets you make those all-important back-up copies without leaving the friendly menu-driven environment you've come to know and love. In the event of a power failure, it rebuilds the data base you were working on.

Our manual is so easy to understand we've also released it as a book. If you want to learn more about data base management in general and Data Design in particular, ask for *Data Base Management Made Easy* at your local book or computer store. At $16.95, it's almost as good a bargain as Data Design itself.

Easier done than said.

Making a data base management system easy for non-programmers made it harder for us, but we think you'd rather use a language you already know than learn another. Since explaining it takes longer than using it, we'd like you to try it at our expense. Send us $225 (or $700, if you insist) and we'll send you Data Design. If, during the first 30 days, you decide you can manage without it, we'll refund your money. Without our software, we think you'll need it.

We put a high price tag on your time, not on our software. To find out more, call us at 503/244-4181 or stop by your local computer store.

Data Design works with a 128K IBM PC with double-sided drives and supports hard disks.

insoft®

10175 SW Barbur Blvd., Suite 202B
Portland, Oregon 97219

THE BASIC SOLUTION

BY JOE JUHASZ

One of the nicest features of the IBM Personal Computer is the high-level graphics language that comes built into its Advanced Basic. On most earlier systems, from the Apple to the million-dollar mainframes from IBM, graphics programs had to be written in assembler. This was a very difficult, time-consuming task, and those who succeeded at it should be commended.

The Personal Computer has made graphics programming available to the novice programmer. Even beginners can create graphs and drawings with the help of simple BasicA commands, such as *color*, *line*, *circle*, *pset*, *paint*, and *draw*. And once drawings have been created, they can easily be manipulated by means of two additional commands, *get* and *put*. *Get* and *put* are highlighted in our program this month.

Briefly, the *get* command grabs a graphics image on the screen and stores it in an array. The *put* command does the reverse; it re-creates a graphics image on-screen from data stored in an array. By using these powerful commands, you can easily create some fancy special effects and animation. Check the IBM Basic manual for the details about *get* and *put*.

The Z-float routine listed in this month's column shows us a simple technique for achieving animation by means of *get* and *put* commands. Our example features an arrow moving across the screen to hit an apple. The program operates by repeatedly erasing an image from one location and redrawing it at another.

When you're attempting to create smooth animation, speed is critical. Therefore, the code shown here, if compiled, would produce much more pleasing results. Without a compiler, you can get some speed improvements by eliminating comments and combining lines 5440 through 5470.

Here's a run-down of the major sections of the program.

Lines 5040 through 5090 initialize variables and set us up for graphics work.

Lines 5120 through 5160 place some images on the screen for us to work with.

Lines 5180 through 5240 provide what the float routine must know: what to move, where to move it, and how quickly to move it. The image to be floated is defined by the coordinates of the upper left and lower right corners of an imaginary rectangle that surrounds the image. X1, Y1 defines the upper left corner; X2,Y2 defines the lower right corner; and X3,Y3 specify the final location of the upper left corner of the image. The variable Float.speed can be set to values from 0 through 10 to vary the speed of the float.

Line 5270 gets the image to be floated from the screen and stores it in the array Pic.

Lines 5290 through 5370 do the calculations to determine the path that will be followed by the floating image. First we calculate the straight-line distance to be floated (*dist = sqr (dx*dx + dy*dy)*). A delay time is calculated, based on the value of Float.speed. Next we determine the number of times the image is to be shown during its float from starting point to final destination. Finally, we calculate the x and y components of the distance the image is to be moved for each step.

You can experiment with the Float.speed variable, the value 100 in line 5330, and the value 10 in line 5340, to see how these numbers affect the outcome of the program.

Lines 5390 to 5480 are the heart of the float routine. These are the lines that repeatedly draw and erase the image at different locations between the starting and ending points.

Line 5430 uses the Basic sound command to cause delays in the float. For maximum speed this line may be removed. *Sound 32767,18*delay!* causes an inaudible sound of Delay! seconds. *Sound 32767,1* causes the program to wait until the prior sound is finished, thereby achieving a delay of Delay! seconds.

Line 5440 erases the image. When you see the *put* command to place an image on top of itself, you cause the image to be erased.

Line 5450 determines the next stop in our float process. As you see, X!, F.dx!, Y!, F.dy! are single-precision variables. This is to ensure the most accurate float possible.

Line 5460 places the image at its next position.

Lines 5490 to 5510 place the image at its final location.

Be careful to code valid values for X1, Y1, X2, Y2, X3, and Y3. Incorrect values may cause errors to occur; these errors are most likely caused by attempts to place the image off the screen.

```
5000 '                    **************************
5010 '                    ***      Z-FLOAT       ***
5020 '                    **************************
5030 '
5040 '**---initialization---**
5050 DEFINT A-Z                                      '-make variables integer
5060 KEY OFF: FOR I=1 TO 10: KEY I,"": NEXT          '-turn off function keys
5070 GOSUB 6280                                      '-switch to color graphics adpt.
5080 SCREEN 1,0                                      '-set medium res. graphics mode
5090 DIM PIC(800)                                    '-dimension array to hold image
5100 '
5110 '
5120 '**---place images on screen for demo---**
5130 GOSUB 6130                                      '-read in ApWorm
5140 CLS
5150 X=50: Y=50: GOSUB 6000                          '-draw an arrow at (50,50)
5160 PUT(260,150),APWORM%                            '-draw ApWorm at (280,150)
5170 '
5180 '**---set float speed from 0 (slow) to 10 (fast)---**
5190 FLOAT.SPEED=9
5200 '
5210 '**---set X1,Y1  X2,Y2  X3,Y3---**
5220 X1=49:  Y1=44
5230 X2=70:  Y2=56
5240 X3=250: Y3=160
5250 '
5260 '**---get the image into array PIC---**
5270 GET (X1,Y1)-(X2,Y2),PIC
5280 '
5290 '**---do calculations to determine movement path---**
5300 DX=X3-X1                      '-DX            = distance from X1 to X3
5310 DY=Y3-Y1                      '-DY            = distance from Y1 to Y3
5320 DIST=SQR(DX*DX+DY*DY)         '-DIST          = distance from X1,Y1 to X3,Y3
5330 DELAY!=(10-FLOAT.SPEED)/100   '-DELAY!        = delay time using FLOAT.SPEED
5340 FLOAT.STEPS=DIST/10           '-FLOAT.steps   = number of steps during float
5350 F.DX!=DX/FLOAT.STEPS)         '-F.DX!         = x distance for each step
5360 F.DY!=(DY/FLOAT.STEPS)        '-F.DY!         = y distance for each step
5370 X!=X1: Y!=Y1
5380 '
5390 '**---cause image to 'float' across screen by   ---**
5400 '**---repeatedly drawing it at new location     ---**
5410 '**---and erasing it from the old location      ---**
5420 FOR I=1 TO FLOAT.STEPS
5430   IF DELAY!>0 THEN SOUND 32767,18*DELAY!: SOUND 32767,1   '-delay
5440   PUT (X!,Y!),PIC                               '-erase previous image
5450   X!=X!+F.DX!: Y!=Y!+F.DY!                      '-determine new location's X,Y
5460   PUT (X!,Y!),PIC                               '-place image at new location
5470 NEXT
5480 '
5490 '**---now place at final location---**
5500 PUT (X!,Y!),PIC                                 '-erase previous image
5510 PUT (X3,Y3),PIC                                 '-place image at final location
5520 '
5530 '**---END---**
5540 END
6000 '***--- draw an arrow ---***
6010   LINE (X,Y)-(X+20,Y),1
6020   LINE (X,Y)-(X+5,Y),2
6030   LINE (X+20,Y)-(X+14,Y-6),1
6040   LINE (X+20,Y)-(X+14,Y+6),1
6050   LINE (X+5,Y)-(X+3,Y-2),2
6060   LINE (X+3,Y)-(X+1,Y-2),2
6070   LINE (X+1,Y)-(X-1,Y-2),2
6080   LINE (X+5,Y)-(X+3,Y+2),2
6090   LINE (X+3,Y)-(X+1,Y+2),2
6100   LINE (X+1,Y)-(X-1,Y+2),2
6110 RETURN
6120 '***--- read in ApWorm graphics symbol ---***
6130 DIM APWORM%(175)
6140 FOR I=0 TO 175: READ APWORM%(I): NEXT
6150 DATA 82, 31,0,0,0,0,0,0,21765,64,0,0,0,5376,20565,0,0,0,0,21845,84
6160 DATA -16381,0,0,21761,21845,-1024,0,0,0,21845,3925,0,0,0,5376,21589
6170 DATA 48,0,0,0,0,-16380,0,0,0,2560,936,0,0,0,0,-21974,3,-32726,0,0,-22016
6180 DATA -21590,-22014,160,0,512,-21846,-21845,-22358,0,0,-22006,-21846,-21846
6190 DATA 168,0,2560,-21846,-21846,-21846,128,0,-22006,-21846,-21846,-32598
6200 DATA 20481,2560,-21846,-21846,-21846,1408,84,-22006,-21846,-21846,-32598,21765
6210 DATA 2560,-21846,-21846,-21846,1408,16453,-22006,-21846,-23126,-32679,21845
6220 DATA 2624,-21846,-21846,26021,21889,16469,-22006,-21846,-23126,31403,21845
6230 DATA 512,-21846,-21846,21925,21845,84,-22014,-21846,-21846,21845,84,512
6240 DATA -21846,-21846,21930,20565,0,-22016,-21846,-21846,21861,64,0,-21974,-21846
6250 DATA -24150,84,0,10752,-21846,-21846,128,0,0,-22006,-21846,-32598,0,0,512
6260 DATA -21846,-21846,128,0,0,-22016,10920,-32598,0,0,0,-32728,-22014,0,0,0,0,0,0
6270 RETURN
6280 '*** Switch to Color Graphics Adapter ***
6290 DEF SEG=&HB800
6300 POKE 0,&H33: COLGRAF=(PEEK(0)=&H33)
6310 POKE 0,&H55: COLGRAF=(PEEK(0)=&H55) AND COLGRAF
6320 IF NOT COLGRAF THEN CLS: LOCATE 12,10: PRINT " Need Color Graphics Adapter ": END
6330 DEF SEG=0: I=PEEK(&H410): POKE &H410,(I AND &HCF) OR &H20
6340 SCREEN 0: SCREEN 1,0: WIDTH 40: LOCATE ,,0,6,7
6350 COLOR 17,1: CLS
6360 RETURN
```

▲

marketalk news

Unless otherwise indicated, software listed runs in DOS on machines with either display adapter and requires 64K and at least one disk drive.

Δ A serial version of the model 130 daisy-wheel printer is available from **Transtar** (Box C-96975, Bellevue, WA 98009; 206-454-9250). Compatible with all major word processors using existing Diablo routines. $950. Δ A bidirectional tractor for the 130 series of daisy-wheel printers has been announced. Fits all Transtar 130 parallel and serial printers manufactured to date. $149.

Δ Learn typing skills in a video-space-war-game format with *MasterType* from **Lightning Software** (Box 11725, Palo Alto, CA 94036; 415-327-3820). Program features seventeen progressive lessons, from introduction of keys to custom spelling and vocabulary games. $49.95.

Δ If you can remember key words easier than keystrokes, you can use *VoiceDrive*, a voice-recognition software interface that comes with *ScratchPad*, the voice-controlled spreadsheet. Both from **SuperSoft** (1713 Neil Street, Box 1628, Champaign, IL 61820; 217-359-2112). Requires 128K and Tecmar voice recognition card. $495. Bundled with card, $995. Δ A Fortran compiler has been released with a full ANSI-66 standard with extensions. Floating-point support is available as an option. $425. Δ *Basic Tutor* is a self-instruction software course that contains graphic illustrations, frequent summaries, and a supplemental manual. $99.

Δ *Edlin Recovery Utility* permits you to recover a file lost during an Edlin session. $35. Δ *DiskPak* is a collection of five utilities including an ASCII examine routine, a hex dump, byte modifier, and more. $60. Δ *BigBuf* is a keyboard extension utility that allows you to increase your keyboard buffer. $45.

Δ Sixteen different creatures, bearing such weapons as the stun wand, the vibra knife, and the disruptor, confront each other on a variety of battlefields in *Galactic Gladiators*, a strategy game from **Strategic Simulations** (883 Stierlin Road, Mountain View, CA 94043; 415-964-1353). You send them on quests where one team must win four or five battles to reach a goal. Requires color/graphics adapter. $39.95.

Δ The MicronEye is a complete vision system that transmits images into the computer's memory, enabling graphics display, image analysis, image storing to disk, and so on. From **Micron Technology** (2805 East Columbia Road, Boise, ID 83706; 208-383-4000). Applications include program animation, security, automated process control, and digitizing. Capable of 256-by-188 resolution and speeds up to fifteen frames per second. $295.

Δ **Transparent Data Systems** (15066 Los Gatos-Almaden Road, Los Gatos, CA 95030; 408-559-0288) has announced a business graphics utility that can draw analytical graphs from existing files without additional programming. *uGraf* can draw bar, pie, line, scatter, and surface graphs from many different sources—databases, spreadsheets, or files. $495.

Δ Borrow programs from *The Electronic Bookshelf* (Box 1409, Norcross, GA 30071), a member-owned collection of personal software. Write for list of programs available and membership application. Membership is free.

Δ The *International Software Databank* is an on-line software supermarket designed to help users find the programs that fit their needs. Recreational, business, industrial, and scientific packages are listed; search your heart out. Contact **Information Research** (10367 Paw Paw Lake Drive, Mattawan, MI 49071; 616-668-2049). Search is $60 per hour.

Δ A low-cost touch tablet called the Koalapad lets you draw directly on the screen with a finger or stylus, bypassing the keyboard. Serves as a sketch pad, custom keyboard with overlays, or a controller for strategy games. From **Koala Technologies** (4962 El Camino Real, Los Altos, CA 94002; 415-964-2992). Software includes educational games, cartoons, and graphics tool kit. $135. Software: $50.

Δ The *Swine Management Series One* program from **Harris Technical Systems** (Box 80837, Lincoln, NE 68501; 402-476-2811) provides data management for hog producers. Includes swine ration analysis, feeder pig analysis, hog selling decisions, and more. $170.

Δ **Queue** (5 Chapel Hill Drive, Fairfield, CT 06432; 203-335-0908) has announced *Queue Catalog #14*, a guide to educational software. Arranged by subject, the catalog covers software for language arts, social studies, physics, foreign languages, and twenty more areas. Specify grade level and computer when writing. Catalog is free.

Δ *Math Blaster* contains over six hundred problems in addition, multiplication, fractions, and so on, for kids ages six through twelve. From **Davidson and Associates** (6069 Groveoak Place, Rancho Palos Verdes, CA; 213-378-7826). The program is an instructional tool in the form of a math arcade game. $49.95.

Δ An RGB interface for use with Sony Profeel monitors provides a sixteen-color display and supports eighty-by-twenty-five-character text. From **Telemax** (780 Lorraine Drive, Box 339, Warrington, PA 18976; 215-343-3000). Includes cable. $99.

Δ Portable storage for hundreds of disks is possible with the Diskette File Tray Caddy, a two-tiered chrome rack on casters from **Ring King Visibles** (215 West Second Street, Muscatine, IA 52761; 319-263-8144). Eight plastic storage trays included. $129.95.

Δ The *Micro/Scan* version of the *Merrill Lynch Equity Research Database* is available from **ISYS** (Box 214, Cambridge, MA 02238; 617-491-6221). Software includes fifty-two Merrill Lynch variables for 1,400 stocks. Trial subscription: $35.

Δ Fleet managers can calculate the anticipated revenue and expense of proposed tanker and dry cargo voyages with the *Voyage Estimator* from **Marine Management Systems** (102 Hamilton Avenue, Stamford, CT 06902; 203-327-6404). Port and sea time, fuel consumption and cost, cargo revenues, loading expenses, and so on can be figured. Deadweight checks and bunker calculations can be made. Requires 256K. $3,000.

Δ Sixteen-year-old Timothy Orr Knight has written *The World Connection*, a book that explores modems, dial-up services, and bulletin boards. Aimed at both the novice and advanced computerist, the book explains technical concepts and applications in a clear style. Published by Howard W. Sams (Box 7092, Indianapolis, IN 46206; 317-298-5400). $9.95.

Δ Add color and custom default parameters to *WordStar* with *SuperStar* from **Relational Solutions** (8723 Woodleigh Drive, Houston, TX 77083; 713-530-4161). Supports reverse video on monochrome. Requires version 3.2 or later. $29.95.

Δ The Easiboard combines the functions of eight separate prod-

PCnet and the IBM-XT

Your Total Solution for IBM PC Local Area Networks.

IBM-PC IBM-PC IBM-XT IBM-PC

Share Your IBM-XT Hard Disk or other Hard Disk (*)

PCnet Major Features

- **Local Area Network and Shared Resource Management**
  - Broadcast Coaxial Bus Network
  - Up to 16 PC's per shared PC Server
  - Up to 8 PC Servers
  - Network errors handled similar to DOS
  - PC to PC communications

- **Computer Vendor Independence**
  - IBM PC, IBM PC-XT
  - Compaq
  - Colby
  - Columbia
  - Corona

- ***Hard Disk Vendor Independence**
  - IBM
  - Santa Clara Systems
  - Tallgrass
  - Tecmar
  - Microcode
  - QCS
  - Datamac
  - Davong
  - Corvus

- **Remote PC Execution**
  - Background program execution at Shared PC
  - Multi-tasking use of Printers, Modems, etc. . .

- **IBM MS-DOS Transparency**
  - MS-DOS 1.1,2.0
  - Off the Shelf program execution over network
  - File Transfers use MS-DOS commands

- **Disk and File Sharing**
  - Public and Private Files/Volumes and Disk
  - File/Record Locking and Unlocking
  - No Dedicated File Server; PC can act as File Server
  - Multiple File Servers

- **Performance**
  - Serial coaxial bus at 1 megabit up to 7000 feet
  - No significant User PC degradation

Future Features

- Networking
  - SNA, ETHERNET,X.25
- Other Computer
  - TI, Victor, Eagles
- Value Added Applications
  - DBMS

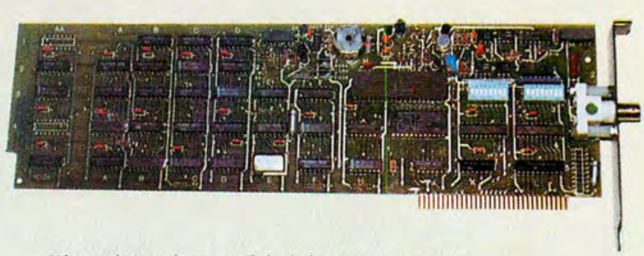
- **Ease Of Use**
  - Single PC slot for PCnet adapter card
  - Standard coax and BNC connectors
  - Automatic Load using AUTOEXEC.BAT

ORCHID TECHNOLOGY

487 Sinclair Frontage Road,
Milpitas, California 95035 • (408) 942-8660
Telex-171596

IBM is a registered trademark of International Business Machines Corp.
MS DOS is a registered trademark of Microsoft Corp.
PCnet is a trademark of Orchid Technology.

Specifications subject to change without prior notice.



When ordering, please specify both the computer/operating system and the hard disk/version. Orchid supports many different combinations and is adding new configuarations each day.

ucts on one plug-in board. From **Easitech** (2215 Perimeter Park, Atlanta, GA 30341; 404-452-7576). The board contains a sorter, a spooler, three disk emulators, a clock and calendar, a printer-switching mechanism, and necessary software. Expandable to 256K. $325 to $595.

Δ A hard-disk subsystem called the Targa II is available in three different models, five to fifteen megabytes. Fifteen-megabyte model has room for five additional expansion cards. From **CMC International** (1720 130th Avenue N.E., Bellevue, WA 98005; 206-885-1600). Units are color- and design-compatible with the pc. $1,490 to $2,445.

Δ Math software from **Sterling Swift Publishing** (7901 South IH-35, Austin, TX 78744; 512-282-6840): *Elementary Mathematics Classroom Learning System—Whole Numbers* and *Fractions/Decimals.* Six disk packages include management and games disks. $495 each. Δ *Discover Basic: Problem Solving with Computers.* $74.95. Δ *How To Program in Basic.* $69.

Δ A single salesperson can generate up to 450 customized sales and follow-up letters daily with *Market Fax*, client and prospect database and marketing support software from **Scientific Marketing** (3303 Harbor Boulevard, Costa Mesa, CA 92626; 714-957-0225). Letters can be personalized and changed to suit client's needs. Requires two disk drives. $495.

Δ Speech-synthesis software can add audible speech to your programs. *PC Parrot*, from **Dragon Data Systems** (1068 Homer Street, Vancouver, British Columbia, Canada V6B 4W9; 604-255-0584), comes complete with utilities. Record your own sound or use library files on disk. No extra hardware required. $39.95.

Δ *Sequitur* is a relational database management system with integrated word processing, report and form generators, and text-merging. From **Pacific Software** (Tenth and Parker, Berkeley, CA 94710; 415-540-5000). The product has been scaled down from a minicomputer environment. No programming skills required. For PC-DOS and other MS-DOS systems. $795.

Δ Enhanced 2780/3780 BSC protocol emulation capabilities are available with *MicroGate*, from **Gateway Microsystems** (Box 10998, Austin, TX 78766; 512-250-9795). The communications software includes a serial communications controller that is installed inside the pc. Requires modem. $895.

Δ A software version of a hardware module that protects software from being pirated, called *The Key*, uses inquiry/response pairs to enhance security. From **Staff Computer Technology** (10457 J Roselle Street, San Diego, CA 92121; 619-453-0303). $75.

Δ Two integrated components to *Micro Cookbook* from **Virtual Combinatics** (Box 755, Rockport, MA 01966; 617-546-6553): *Desserts* is an optional recipe disk. $12. Δ The *Advanced Function Package* includes expanded search capabilities, multiple-direction recipe screens, sound, and recipe transfer between disks. Requires *Micro Cookbook*. $30.

Δ A programmable 3-D spreadsheet with a built-in language from **Datamension** (615 Academy Drive, Northbrook, IL 60062; 312-564-5060) lets the user save pages one at a time. *Report Manager* also automatically expands to take advantage of all available RAM and features single-keystroke commands. $399.

Δ *Diskette Manager* is a library program with label printing capability. A catalog file is constructed with the basic label information. From **Lassen Software** (Box 1190, Chico, CA 95927; 916-891-6957). Sixty-four file names can be printed on one four-by-one-inch label. $50.

Δ As many as one hundred stocks can be entered into the *Stock Portfolio Reporter* from **Micro Investment Systems** (Box 8599, Atlanta, GA 30306; 404-892-3194). Market price updates can be done automatically using Dow Jones News/Retrieval; P/E multiples, dividend yields, and gain or loss amounts can be calculated. Requires 128K, two disk drives, eighty columns, printer, modem, and subscription to Dow Jones. $179.

Δ *Qbase* is a personal database and report-writing package from **Applied Software Technology** (14125 Capri Drive, Los Gatos, CA 95030; 408-370-2662). Includes automatic checklists, extensive data-entry checking, an embedded calculator, and more. $189. Δ *Purchase Order* is a *VersaForm* template designed for small or large purchasing departments. $49.95. Δ *Legal Office Manager* is a client billing and scheduling system. $249.

Δ A videotape learning presentation called *The VisiCalc Program* can teach users how to set up and implement the spreadsheet program in one hour. From **Micro Learning Concepts** (120 East Fifty-sixth Street, New York, NY 10022; 212-980-3552). Reference guide to commands included. $129.95.

Δ *ReadiTerm* is a communications package that gives you access to a mainframe as well as to the Source and Dow Jones. From **ReadiWare Systems** (Box 680, West Redding, CT 06896; 203-431-3521). It can issue predefined commands, capture the output, and disconnect for off-line viewing. $50.

Δ You can have up to nine jobs running at the same time with *Multi-Job* from **B&L Computer Consultants** (Box 4323, Boise, ID 83704; 208-377-8088). Jobs can be ordinary DOS or Basic programs. DOS 1.1 required; 128K suggested. $159.

Δ An application development module, *dBase II RunTime*, enables programmers to market specialized applications written and developed in *dBase II*. Licensing agreement included. From **Ashton-Tate** (9929 West Jefferson Boulevard, Culver City, CA 90230; 213-204-5570). $500. Δ *Friday!* is a database management system that allows you to create and change files from anywhere in the system. A math field allows you to create rules to govern calculations you want performed. Includes mailing label function. $295.

Δ Customize *Multiplan* worksheets by automatically entering formulas and formatting the sheets, using one of the *Multi-Tool* packages from **Microsoft** (10700 Northup Way, Bellevue, WA 98004; 206-828-8080). The *Budget* system is for budget planning and control. $150. Δ The *Financial Statement* system is for performing financial statement ratio analysis. $100. Δ The SystemCard integrates four functions, including serial and parallel interfaces and clock/calendar with battery backup. Expandable to 256K. $475 to $995.

Δ The Multiple Adapter Interface board from **Amdek** (2201 Lively Boulevard, Elk Grove Village, IL 60007; 312-364-1180) contains 128K RAM, a two-position DIP switch for configuring mono or color applications, a hi-res light pen circuit, and more. $799.

Δ A full-color training catalog featuring six audio cassette courses for beginning *VisiCalc*, *WordStar*, and pc users is available from **FlipTrack Learning Systems** (Box 711, Glen Ellyn, IL 60137; 312-790-1117). Catalog is free.

Δ A lightweight, twelve-inch diagonal RGB monitor is available from **NEC Home Electronics** (1401 Estes Avenue, Elk Grove Village, IL 60007; 312-228-5900). The unit features 690 dots horizontal and 230 lines vertical. $798.

Δ A series of line interface adapters from **Data Link** (1085 N.W. Twelfth, Issaqua, WA 98027; 206-392-2232) uses the existing wiring within buildings as a data communications network. Each model has specific applications. All adapters operate in a hunt mode, searching for the first available port in a networked computer. $250 to $525.

Δ Communicate with IBM 3270 controllers over standard phone lines using IrmaLine, emulation firmware from **Technical Analysis** (120 West Wieuca Road N.E., Atlanta, GA 30042; 404-252-1045). Installed at the 3270 site, the interface provides vertical scrolling and other capabilities. $1395. Δ *Irmalette* is an optional companion product, giving remote pc and XT users the same selective data transfer capabilities currently offered in the Irma board. $325.

Δ New publications from **Hayden Books** (50 Essex Street, Rochelle Park, NJ 07662; 201-843-0550): *Secrets of Better Basic*, by Ernest Mau, reveals sophisticated programming tricks and techniques. Softback. $14.95. Δ *Using Microcomputers in Business: A Guide for the Perplexed, Second Edition*, by Stanley S. Veit, is a background reference for purchasers of systems and software for small businesses. Softback. $12.95. Δ *Microcomputers Can Be Kid Stuff*, by Anna Mae Walsh Burke, teaches young people about using micros productively. Softback. $8.95.

Δ First releases in the Vision line from **Bruce & James Program Publishers** (2307 Broadway, New York, NY 10024; 800-531-1309): *WordVision* is a word processor. $49.95. Δ *SpellVision* is the companion spelling checker. $39.95. Δ *PowerPack* is a utilities disk that gives *WordVision* split screen, multiple format, and other options. $49.95.

Δ The ability to send and receive messages or files simultaneously between pcs while, at the same time, printing and editing locally is available with *Relay*, communications software from **VM Personal Computing** (60 East Forty-second Street, New York, NY 10165; 212-697-4747). Comes with full-screen text editor. $140.

Δ Extend the use of *VisiCalc* with *StretchCalc* software, featuring graph- and bar-chart-generating capabilities and sorting and rearrangement of columns and rows. From **Multisoft** (14025 S.W. Farmington Road, Beaverton, OR 97005; 503-626-4727). Command sequences can be stored and invoked with single keystroke. Requires 128K (DOS 1.1) or 192K (DOS 2.0) and color/graphics adapter. Printer with Graftrax option required to print graphics. $99.

Δ Incorporating a hardware curve generator usually found on mainframes, the modular Cono#Graph graphics generator features 256 colors, 640-by-400 resolution, and 128K of graphic memory. From **Conographic** (2268 Golden Circle, Newport Beach, CA 92660; 714-642-6778). The multiple-board professional system provides high-speed drawing of vectors, circles, ellipses, and conic sections. $125 to $895.

Δ Effective as a medium-size document processor as well as a Basic instructional program, the *B.Writer* contains many features found on more advanced word processors. From **Automated Information Systems** (1503 Avenue J, Lubbock, TX 79401; 806-762-6604). $39. Compiled version: $59.

Δ *VisiCalc* spreadsheets can be converted into IFPS models with *Vis-Escape*, from **Clarity Software** (11103 Spicewood Parkway, Austin, TX 78750; 512-258-5473). Only row/column models can be converted. Converted models may be uploaded to IBM TSO and CMS, using other software. Requires DOS 1.1. $600.

Δ **The Programmer's Shop** (135 Main Street, Maynard, MA 01754; 617-897-4750) is a software outlet that publishes a Critical Skills List—a list of available independent programmers looking for interesting assignments. Updated weekly, the list is sent free with every software order. Past lists: $8.50 each.

Δ *Dental Practice Management* is an accounts-receivable billing system for one to nine dentists or hygienists in a practice. From **Digital Marketing** (2363 Boulevard Circle, Walnut Creek, CA 94595; 415-938-2880). The package provides seven reports at the push of a button as well as a mail list/recall system. Requires CP/M-86. $995. Δ *Notebook* is a database management system designed for the storing and retrieving of text. Can select and sort any text in any field. $150.

Δ A programmable text formatter, *Sprinter-2* is designed for large and complex text formatting applications. From **Scenic Computer Systems** (14852 N.E. Thirty-first Circle, Redmond, WA 98052; 206-885-5500). The word processor is geared toward preparation of

Lee McFadden makes learning to use your computer and its programs an easy, rewarding experience.

Mastering a new system or even a new program is not easy. It takes a lot of study, a lot of trial and error, a lot of back-and-forth between dense manuals and your computer. Wouldn't it all be so much easier if you had a knowledgeable friend sitting by your side?

Lee McFadden is that friend. He's been producing widely acclaimed teaching tapes for over ten years. He knows what information you need, and exactly when you need it. Team his skill with the know-how of technical experts, and you get a faster, easier way to master your computer and all its power.

Patented Audio Cassettes

Lee guides you patiently, step-by-step, while you're sitting at your keyboard. No connection between the computer and your cassette player is necessary. You learn by *doing*, instead of by reading. And even the tapes are special—a patented FlipTrack system lets you go at your own pace, covering as much or as little detail as you want.

Ideal for Training

Use the courses over and over again. Share them with friends, family, students and staff. Make "computer literacy" a reality in your local schools. Bring a new employee "up to speed" in just hours, instead of weeks. The fully indexed Guide included with each course is, by itself, an invaluable, on-going reference.

Moneyback Guarantee

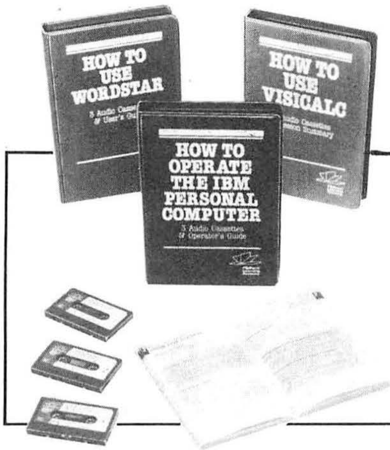
You may order any of the FlipTrack courses from us on a 15-day "right of return" basis. Try a lesson or two yourself. Share them with others. If you're not delighted at how quickly and easily you begin to learn, simply return the program in its original condition for a full refund. No questions asked.

How to Order

Drop by your local dealer for a demonstration. Or Visa and MasterCard holders may add $2.50 shipping and handling to the prices, and order toll-free:

1-800-227-1617, Ext. 439

(In California, call 1-800-772-3545, Ext. 439.)



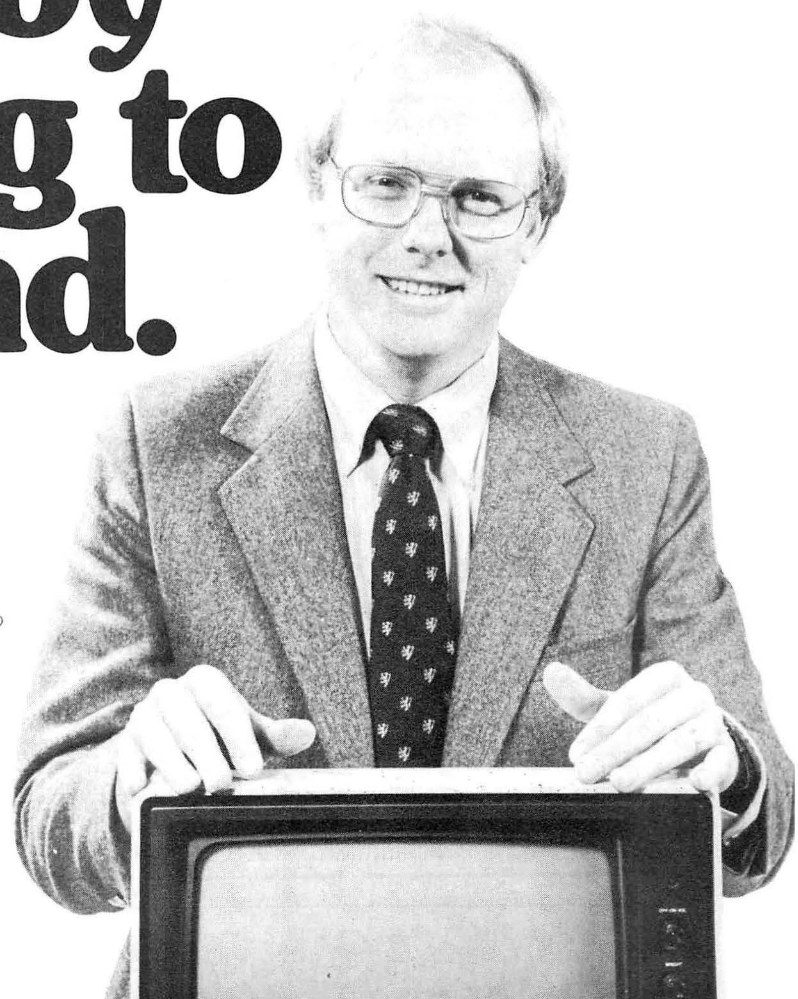
Choose from these rewarding courses:

How to Operate the IBM PC®
3 audio cassettes & Operator's Guide $49.95

How to Use VisiCalc®
4 audio cassettes & Lesson Summary $65.00

How to Use WordStar®
3 audio cassettes & User's Guide $49.95

Learn by listening to a friend.



FlipTrack® Learning Systems

A Division of Mosaic Media, Inc.
526 N. Main St., Dept. SI-C
Post Office Box 711
Glen Ellyn, IL 60137
312-790-1117

reports, books, manuals, scripts, and newsletters. Requires UCSD p-System version IV.0 or later and two disk drives. $350. *Spelling Checker*: $125.

Δ Engineers, architects, and financial planners can model problems with *Cope*, a spreadsheet that comes with cost-estimating, cash-flow, and real estate syndication models. From **Antech** (788 Myrtle Street, Roswell, GA 30075; 404-993-7270). $395.

Δ Logic design and simulation are possible with *Micro-Logic*, from **Spectrum Software** (690 West Fremont Avenue, Sunnyvale, CA 94087; 408-738-4387). Engineers can draw logic diagrams directly on a CRT screen. The netlist required for simulation is generated automatically from the drawn diagram or may be typed in directly. $450.

Δ *Statistician's Mace* calculates statistics frequently used by scientists, business researchers, engineers, and others who need to analyze data from surveys and experiments. From **Mace** (2313 Center Avenue, Madison, WI 53704; 608-241-4566). The package performs the Hodges-Lehman aligned ranks test. $145.

Δ For steady-state systems engineering applications, *S3E System 1* is a turnkey system analysis package composed of nine basic modules and supporting software. Complete set of codes and hardware: $6,560. From **PSI/Systems** (Research Park, Box 3100, Andover, MA 01810; 617-475-9030). Modules can be purchased separately: *ThermoPak* computes various thermodynamic properties. $100. Δ *Thermo3* models either water or ammonia. $200. Δ *MathPak* does numerical analysis. $75. Δ *HxPak* does heat exchange analysis. $200. Δ *S3EPak Module Library* is a tutorial on writing flowsheet analysis programs. $50. Δ *FluidPak* computes condition of fluid handling equipment. $225. Δ *ReactorPak* is for coal gasifier reactor analysis. $325.

Δ Add six slots for additional memory and ports with the Sup'r Extender expansion unit from **M & R Enterprises** (910 George Street, Santa Clara, CA 95050; 408-980-0160). It's half the size of the pc and matches it in appearance. $500.

Δ Test drive Unix by subscribing to The Solution, a Unix timesharing system from **Kersmeyer Electronic Design** (16411 Del Mar Lane, Huntington Beach, CA 92649; 714-846-7739). Some of the languages available are C, SMC Basic, RM Cobol, and SVS Pascal. Programmer's Workbench provides helpful commands. Connect charge as low as $3.95 per hour.

Δ *Max, the Production Manager* is a production planning and control system for small companies and divisions from **Micro MRP** (Century Plaza 1, 1065 Hillsdale Boulevard, Foster City, CA 94404; 415-345-6000). Meeting the needs of manufacturing companies, the package is designed for single-user or multiuser configurations. Hardware included: $25,000.

Δ Custom-designed quilted fabric system unit covers in water-repellent materials are available in gray, black, brown, baby blue, and other colors from **PC Covers** (Box 8286, Foster City, CA 94404; 415-341-0101). Printer, monitor, and keyboard covers also available. $6.95 to $36.95.

Δ SmarTerm/PC model TE400-FT allows you to emulate a Data General D100, D200, or D400 terminal and transfer program and data files at speeds up to 9,600 baud. From **Persoft** (2740 Ski Lane, Madison, WI 53713; 608-233-1000). $125. Δ Version 2.1 of SmarTerm/PC model TE100-FT adds several enhancements to its Digital Equipment VT series emulation. $150.

Δ Engineering software from **Software Systems** (5766 Balcones Drive, Austin, TX 78731; 512-451-8634): *Datafit* performs regression analysis. Requires a printer. $195. Δ *SteamCalc* computes individual values of thermodynamic steam properties. $195. Δ *Combustion* computes combustion efficiency of industrial and commercial boilers and process heaters. $390. Δ *HeatFlo* computes heat losses from pipes and other surfaces. $295. Δ *FluidFlo* computes the pressure loss and power consumption for a series of up to seventy sections of pipe. $265. Δ *Cogen* performs thermodynamic and financial analysis on the benefits of installing a turbine/generator unit. $590. All programs require eighty columns. Source code can be purchased separately.

Δ The split-screen word processor *Pie: Writer* from **Softwest** (465 South Mathilda Avenue, Sunnyvale, CA 94086; 408-737-0624) makes full use of the function keys and allows alternate screen editing. Files are compatible with Apple version of the program. $199.95.

Δ *Professor DOS* is a software tutorial designed to guide the novice as well as the experienced user through fundamental PC-DOS system concepts, commands, and editing functions. From **Individual Software** (24 Spinnaker Place, Redwood City, CA 94065; 415-591-4166). $59.95.

Δ Using data entered at a cash register, *The Retailer* tracks sales, keeps inventory up to date, and prints a variety of reports. From **Retail Solutions** (1211 Alderwood Avenue, Sunnyvale, CA 94086; 408-734-0653). A bar wand or scanner and accounts-receivable module can be added to the system. Requires XT or hard disk, 192K, printer, communications adapter, and mono display. With one cash register: $5,450. Complete hardware, installation, and training package: $15,000.

Δ A new version of the UCSD p-System that runs one and a half to five times faster has been developed by **Network Consulting** (Discovery Park, 3700 Gilmore Way, Burnaby, British Columbia, Canada V5G4M1; 604-430-6448). Interpreter module features fast floating point and long integer support. Complete development kit: $845. Run-time version: $150.

Δ In the arcade game *Big Top*, you maneuver Wendell the acrobat through a many-ring circus, ducking, climbing, and dodging obstacles. $39.95. Δ In *Master Miner*, you must mine the gems of the asteroid belt while protecting yourself from the cosmic claim jumpers. $39.95. Both are new from **Funtastic** (5-12 Wilde Avenue, Drexel Hill, PA 19026; 215-622-5716). ▲

With the Greatest of Ease...

FLYING WITH THE OAG

by JoAnn Levy

"I want you both at that meeting in Boston a week from Monday. Eleven o'clock sharp."

"Right, J.D.," replies Mr. Notwithit who hurries off to jot his secretary a note to make travel arrangements for him on Monday.

"Right, J.D.," says Ms. Withit, glancing at her watch. The Baltimore traffic isn't too bad, and she will be home about six. The standard rates are in effect for both services—$5 an hour for CompuServe time, plus $21 an hour for OAG time.

She logs on to the information service and at the prompt types: **Go OAG 200**. Her screen responds:

```
OAGEE                         PAGE OAG-200

REQUEST RECORDED, ONE MOMENT, PLEASE
CONNECTED TO 120AG

WELCOME TO THE OFFICIAL AIRLINE GUIDE (OAG),
 COPYRIGHT 1983, OFFICIAL AIRLINE GUIDES,
  INC., OAK BROOK, ILLINOIS 60521

   OAG NEWS LAST UPDATED 1 JULY

     PRESS ENTER KEY FOR IMPORTANT NEWS, OR
ENTER /I, /F, /S, /M
```

She types the shortcut request for schedules from Baltimore to Boston: **/S Baltimore;Boston 22 Jul**

She could have entered /S to respond to menu queries, but entering the departure city, a semicolon, destination city, and date takes her directly to the daily airline schedule. The screen displays:

ENTER L#,X#,S,RS (#=LINE NUMBER)

Despite all the ... about airline deregulation, she's still
... for a one-hour round-trip flight—$88 to
... there are limitations for the lower fares
... can qualify for the $88 fare. She types
... for the flight on line 1.

B FRI–22 JUL
S ROUND-TRIP
...ARE CODE BE70X67

...ADVANCE PURCHASE

...ABLE FOR TRAVEL FROM

...IRED IS 1 SUNDAY.
...WED IS 60 DAYS.

DIRECT FLIGHTS
FROM–BALTIMORE,MD,U...
TO–BOSTON,MA,USA

1	430P	BWI	541P...
2	655P	BWI	804P...
3	805P	BWI	913P...
4	815P	BWI	918P...

NO LATER DIRECT F...
ENTER + FOR CONNE...
ENTER –,O,CX,X#,F#,RS...

Why not take that flight ...
Boston in time for dinner wit...
Cape Cod. To see the fares f...
request for line 1.

FARES IN US DOLLARS
SELECTED FOR BWI–AL 240 BOS

#	ONE-WAY	RND-TRP	ARLN/CLA...
NO LOWER FARES IN CATEGORY			
1*		88.00	AL/B
2*		88.00	AL/B
3*		199.00	AL/F
4	109.00	218.00	AL/...
5	136.00	272.00	A...

Photo illustration by Barry Hoff and Kevin McKeon

```
DIRECT FLIGHTS                                  FRI-22 JUL
FROM-BALTIMORE,MD,USA
# TO-BOSTON,MA,USA
NO EARLIER DIRECT FLIGHT SERVICE
1     710A  BWI   818A  BOS  AL 350   D9S   B 0
2     735A  BWI   843A  BOS  DL 266   72S   B 0
3     810A  BWI   913A  BOS  PI 316   73S   B 0
4    1100A  BWI  1208P  BOS  DL 130   72S   S 0
5    1215P  BWI   126P  BOS  AL 214   B11   S 0
6     120P  BWI   231P  BOS  AL 270   B11   S 0
ENTER +,CX,X#,F#,RS                (#=LINE NUMBER)
```

The flight on line 3 would get her to Boston in time for the meeting, but would mean departing Baltimore at 8:10. She hates early morning flights. She enters the + command to scroll through more flights.

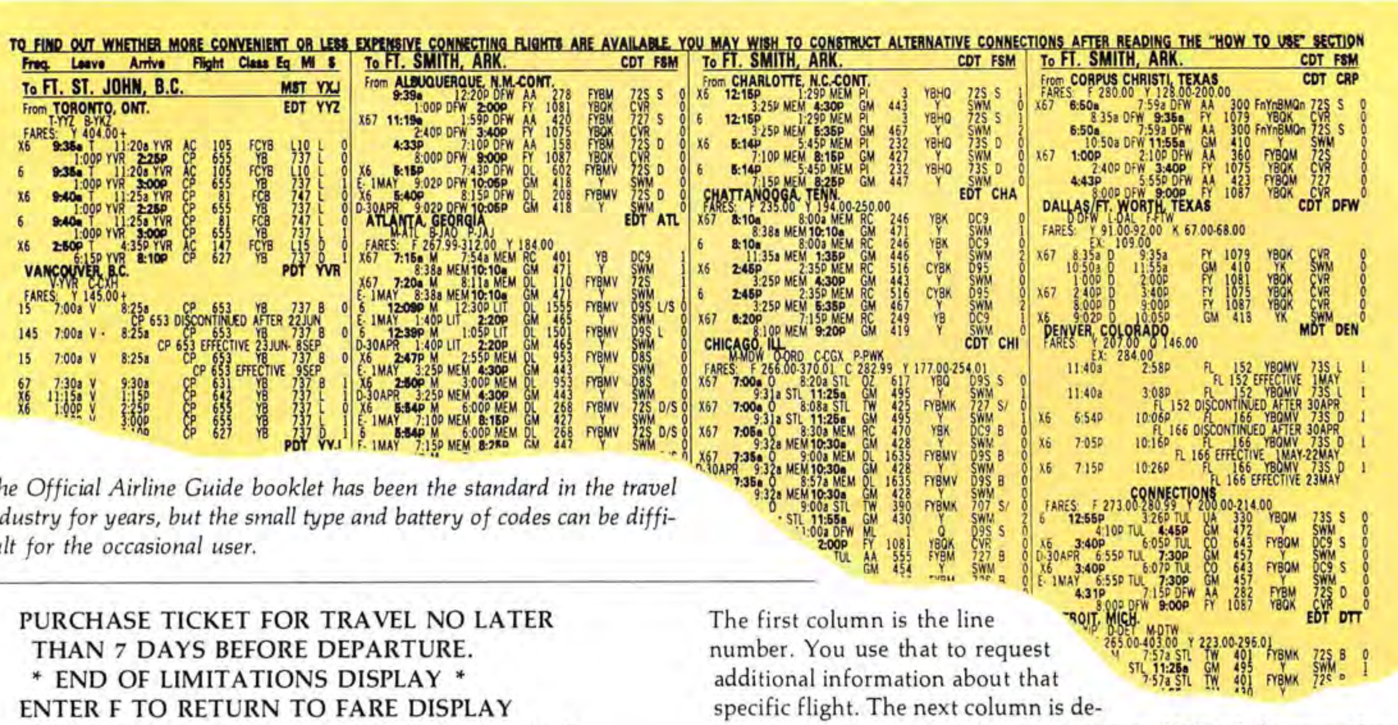
```
DIRECT FLIGHTS                                  FRI-22 JUL
FROM-BALTIMORE,MD,USA
# TO-BOSTON,MA,USA
1     430P  BWI   541P  BOS  AL 240   B11   S 0
2     655P  BWI   804P  BOS  DL1622   D9S   S 0
3     805P  BWI   913P  BOS  AL  48   D9S     0
4     815P  BWI   918P  BOS  PI 484   73S     0
 NO LATER DIRECT FLIGHT SERVICE
ENTER + FOR CONNECTIONS
ENTER -,O,CX,X#,F#,RS              (#=LINE NUMBER)
```

Why not take that flight on line 1 next Friday? She could be in Boston in time for dinner with her sister and spend the weekend on Cape Cod. To see the fares for that flight she enters F1, the fare request for line 1.

```
FARES IN US DOLLARS                             FRI-22 JUL
SELECTED FOR BWI-AL 240 BOS

#  ONE-WAY  RND-TRP  ARLN/CLASS  FARECODE
 NO LOWER FARES IN CATEGORY
1*            88.00    AL/B        BE70X67
2*            88.00    AL/B        BE70Z67
3*           199.00    AL/B        BE77
4  109.00    218.00    AL/M        M
5  136.00    272.00    AL/Y        Y
 NO HIGHER FARES IN CATEGORY
 * ENTER L# TO VIEW LIMITATIONS
ENTER L#,X#,S,RS                   (#=LINE NUMBER)
```

Despite all the publicity about airline deregulation, she's still amazed at the range of fares for a one-hour round-trip flight—$88 to $272. The asterisk tells her there are limitations for the lower fares but, who knows, maybe she can qualify for the $88 fare. She types L1 to view the limitations for the flight on line 1.

```
BWI-BOS AL  CLASS B                             FRI-22 JUL
    88.00 US DOLLARS                          ROUND-TRIP
LIMITATIONS FOR FARE CODE BE70X67

FARE DESCRIPTION: ADVANCE PURCHASE
 EXCURSION FARES
BOOKING CODE: B.

FARE IS ONLY AVAILABLE FOR TRAVEL FROM
 MON THRU FRI.

MINIMUM STAY REQUIRED IS 1 SUNDAY.
MAXIMUM STAY ALLOWED IS 60 DAYS.
```



The Official Airline Guide booklet has been the standard in the travel industry for years, but the small type and battery of codes can be difficult for the occasional user.

PURCHASE TICKET FOR TRAVEL NO LATER
THAN 7 DAYS BEFORE DEPARTURE.
* END OF LIMITATIONS DISPLAY *
ENTER F TO RETURN TO FARE DISPLAY
ENTER S TO RETURN TO SCHEDULE DISPLAY

Ms. Withit types /Q to quit the OAG, logs off CompuServe, contacts her travel agent to make a reservation, and calls her sister.

****

This particular scenario is fictitious—except for the OAG information. But one is tempted to finish the story.

Ms. Withit enjoys a weekend at Cape Cod, arriving relaxed at the eleven o'clock meeting. She smugly submits her $88 expense voucher to J.D., having enjoyed a weekend trip with the flight paid by the company. Mr. Notwithit submits an expense voucher for $218, the only fair available when his secretary and the travel agent finally stopped calling each other.

This is just one example of why you should learn to use the OAG; there are as many others as there are flights and passengers.

The electronic edition of the Official Airline Guide database contains approximately seven hundred thousand flight schedules for 650 airlines serving one hundred five thousand city pairs. It is available through direct subscription, CompuServe, and various other connection services. The database contains schedules for all direct flights and thousands of connecting flights operating throughout the world, and although international fare information currently is unavailable, plans are to provide it sometime next year. Fare information is provided for all direct flights and connections in North America.

The Daring Young Man. Using the OAG is relatively simple, once you're accustomed to reading the fairly cryptic codes. Let's take a moment and figure them out. Look at the first example again:

DIRECT FLIGHTS FRI–22 JUL
FROM–BALTIMORE,MD,USA
TO–BOSTON,MA,USA
NO EARLIER DIRECT FLIGHT SERVICE

1	710A	BWI	818A	BOS	AL 350	D9S	B	0
2	735A	BWI	843A	BOS	DL 266	72S	B	0
3	810A	BWI	913A	BOS	PI 316	73S	B	0
4	1100A	BWI	1208P	BOS	DL 130	72S	S	0
5	1215P	BWI	126P	BOS	AL 214	B11	S	0
6	120P	BWI	231P	BOS	AL 270	B11	S	0

The first column is the line number. You use that to request additional information about that specific flight. The next column is departure time (local), then departure airport. BWI is Baltimore. If you don't recognize the airport designation, you can query it with a preceding '?' and the OAG will identify it for you. The fourth column is arrival time (local) for the destination in the fifth column. The sixth column is the airline—Delta, Piedmont, and so forth—followed by the flight number. The eighth column is the equipment—L1011, DC-10, 727, or whatever. The ninth column is what you get to eat: B, L, D, S—breakfast, lunch, dinner, or snack. The final column is the number of stops en route. If you're looking for a nonstop flight on a direct flight schedule, check this column for a zero.

On the Flying Trapeze. You can access either schedules or fares directly and switch back and forth between them. Our fictitious scenario demonstrated a schedule access. You can use the same "formula" to call up fare information. Suppose you wanted the cheapest fare between Los Angeles and Boston on July 23. After accessing the OAG, you would type:

/F LAX; Boston 23 Jul

You can use full city names, airport designations, or combinations of the two. The display accessed by the above request is:

FARE MENU

FARES FOR DIRECT FLIGHTS AND CONNECTIONS		FARES FOR DIRECT FLIGHTS ONLY
--		---
1	COACH CLASS AND EQUIVALENT FARES	6
2	FIRST CLASS AND EQUIVALENT FARES	7
3	BOTH COACH AND FIRST CLASS FARES	8
4	ADVANCE-PURCH AND EXCURSION FARES	9
5	ALL OF THE ABOVE FARES	10

PLEASE ENTER A NUMBER

For the sake of the example, we're interested in all fares for direct flights only, so we enter 10.

FARES IN US DOLLARS SAT–23 JUL
SELECTED FOR LAX–BOS

```
#  ONE-WAY  RND-TRP  ARLN/CLASS  FARECODE
 NO LOWER FARES IN CATEGORY
1*           298.00   NW/B        BE77
2*           318.00   AA/B        BE77
3*           318.00   CO/M        ME77K
4*           318.00   TW/B        BHE77
5*           318.00   UA/B        BE77
6*           378.00   NW/B        BE77Q
7*           379.00   NW/B        BE70
8*           398.00   CO/M        ME77
 * ENTER L# TO VIEW LIMITATIONS
ENTER +,L#,X#,S#,R#,M,RF(#=LINE NUMBER)
```

The message in the heading tells us there is no lower fare than that on line 1 and the asterisk signals limitations. To see what they are, we type L1.

```
LAX-BOS NW CLASS B                     SAT-23 JUL
  298.00 US DOLLARS                   ROUND-TRIP
LIMITATIONS FOR FARE CODE BE77

FARE DESCRIPTION: ADVANCE PURCHASE
 EXCURSION FARES
BOOKING CODE: B.

MINIMUM STAY REQUIRED IS 7 DAYS.
MAXIMUM STAY ALLOWED IS 14 DAYS.

PURCHASE TICKET FOR TRAVEL NO LATER
 THAN 7 DAYS BEFORE DEPARTURE.
 * END OF LIMITATIONS DISPLAY *
ENTER S TO VIEW SELECTED NW SCHEDULES
ENTER F TO RETURN TO FARES DISPLAY
```

Since we're going to Boston for a convention and a few days' recreation, we can meet the minimum stay and other requirements. Now we want to see what the schedule is for this fare. If you think it's the infamous "red-eye," you're wrong. We enter S.

```
                 DIRECT FLIGHTS        SAT-23 JUL
SELECTED FROM NW FARE DISPLAY
FROM-LOS ANGELES,CA,USA
# TO-BOSTON,MA,USA
 NO EARLIER DIRECT FLIGHT SERVICE
1  1100A  LAX  835P  BOS  NW  166  D10  L  1
 NO LATER DIRECT FLIGHT SERVICE
ENTER R FOR RETURN NW SCHEDULES
ENTER X#,L,R,F,RF                 (#=LINE NUMBER)
```

To expand the code into plain English, we type X and the line number, 1.

```
  EXPANDED DIRECT FLIGHT DISPLAY
LEAVE-11:00A       ON-23 JUL
FROM-LOS ANGELES,CA,USA
NORTHWEST ORIENT AIRLINES FLIGHT 166
AIRCRAFT-MCDONNELL DOUGLAS DC10
CLASS-FIRST/COACH/ECONOMY
MEAL-LUNCH/DINNER
ARRIVE-8:35 P      ON-23 JUL
AT-BOSTON,MA,USA
ELAPSED TRAVEL TIME  6H 35M

ENTER S TO RETURN TO SCHEDULES
ENTER F FOR SELECTED FARES
```

Do-It-Yourself Specials. Why not just let your travel agent look it up? There are good reasons, one of which is that by doing it yourself you know for sure you've got the cheapest flight, the agent doesn't have to call you back and ask if an 11 o'clock departure is okay, you don't have to discuss it with your spouse and then call the agent, and so forth and so forth. And there are numerous possibilities your travel agent has no interest in telling you about, and you have no way of asking about.

Let's see what we get when we look at connections. From the last display, we type S to return to schedules, and then type CX.

```
                CONNECTIONS                 SAT-23 JUL
FROM-LOS ANGELES,CA,USA
# TO-BOSTON,MA,USA
1    1000A  LAX   242P  DFW  DL1140    L10  L  0
      342P  DFW   810P  BOS  DL 824    767  D  0
2    1000A  LAX   603P  JFK  AA  40    747  L  0
      915P  JFK  1014P  BOS  AA 118    72S     0
3    1005A  LAX   334P  STL  TW  78    707  L  0
      430P  STL   800P  BOS  TW 150    72S  D  0

ENTER +,-,DF,X#,F#,RS              (#=LINE NUMBER)
```

The flights on lines 1 and 3 aren't too appealing. You have no interest in spending an hour in the Dallas-Fort Worth Airport, or in Saint Louis. But the flight on line 2 goes to JFK. Isn't that the other New York airport? You're unsure, so you type: ? JFK.

RESPONSE TO YOUR HELP REQUEST
JFK=
NEW YORK,NY,USA/J.F. KENNEDY

Good. Now, the American Airlines flight on line 2 gets into JFK at 5:55 p.m. and requires a plane change for the connection to Boston. But: you have good friends in New York who'd love to meet you at the airport for dinner and your stopover of three hours would permit a nice visit. You get into Boston at 10:12 p.m. instead of 8:35 p.m. and the flight costs $20 extra. But maybe it's worth it to you. The point is, only you know whether you can take advantage of various schedules. When you ask travel agents for the cheapest flight to Boston, they aren't going to ask if you'd like to spend another $20 and have dinner in New York.

See the USA. The possibilities for creative scheduling are limitless when you have access to all the schedules and fares yourself. Suppose you see an ad for a cheapie flight to Houston. You may not want to go to Houston, but perhaps you can launch yourself inexpensively from there to a destination you favor. Or, suppose you take a direct flight from Los Angeles to Chicago for business that concludes on a Friday and you don't need to be back to L.A. until Monday. Where might you spend the weekend? Just call up the connecting schedules for flights from Chicago to Los Angeles and find out.

Easy Street. Using OAG's commands to see connecting flights, fares, and so on is quick and simple. Each display, as you can see from the examples, includes the menu options available from that particular display. Take a couple of minutes to look at the accompanying list of OAG command explanations and the display examples. You should be able to figure out exactly what information you can access from the various displays.

Also, you can get a narrative explanation of the commands, at CompuServe rates (no surcharge), by typing Go OAG and selecting the appropriate menu option. Further, there's an OAG Help Desk with a toll-free number: (800) 323-4000.

The Bottom Line. Yes, there's a surcharge for using the OAG. As mentioned in the opening scenario, the standard rate (after 6 p.m. and weekends) is $21 an hour, plus the standard CompuServe service rate. Prime time (weekdays, 8 a.m. to 6 p.m.) charges are $32 an hour, on top of the information service prime rates. Using the OAG through CompuServe during prime time works out to about ninety cents a minute; standard time at about forty-five cents a minute. Our Ms. Withit might have spent a couple of dollars for the few frames of information she required. On the other hand, she saved her department's travel budget more than a hundred dollars. That's a pretty good return on an investment of two bucks.

If you expect to use the service frequently, you may be interested in a direct subscription, rather than accessing through CompuServe. There's a one-time charge of fifty dollars and usage is billed in units, with a unit equal to ten cents for one minute; additionally, every schedule screen display costs two units, every fare display three.

Better Than Kansas City. OAG fares are updated daily, schedules weekly. Approximately one hundred fifty airlines worldwide have direct access to the OAG database and update their own schedule and fare information continuously. British Airways, for example, has terminals in both New York and London with the capability of inputting to the database in real-time mode.

OAG, a wholly owned subsidiary of Dun and Bradstreet, processes new information constantly at its headquarters in Oak Brook, Illinois. Although the database is down between 4:30 and 5:00 a.m. Central Time for updating, according to Dave Shaffer, OAG's senior vice president of electronic publishing, "updating only takes about five minutes."

Shaffer is one of the forty to fifty of OAG's employees "dedicated" to the electronic edition. The design of the electronic edition, which has been under development for the past three years, was not, says Shaffer, to have a separate division; almost all nine hundred OAG employees have input and responsibility.

The print edition of the OAG dates from 1929, when the company was founded, but the electronic edition has only been on-line since May 1. Its prototype ran until February and certain design features are still being incorporated, such as randomization of competing airlines with identical schedule and/or fares. You won't always see American Airlines preceding United.

You, Me, and ITT. Who uses the OAG? Of the one hundred ten thousand present subscribers (anticipated at two hundred thousand by September)—three thousand direct subscribers plus those who access through various database services—65 percent are business accounts. Only 35 percent are travel agents and airlines. The friendly OAG EE was designed for the untrained, unknowledgeable user, not for the travel professional.

In addition to friendliness, there's a "fair deal" for the traveler that was previously unavailable. Even if your travel agent wants to secure you the cheapest flight, it may not be possible. Ninety percent of the agents use automated systems, it's true, but most are provided by a sponsoring airline. The implications are fairly apparent.

You do, of course, still need an agent for availability information and booking. But at least when it comes to fares and schedules, you've got a choice. Lots of choices. ALL the choices.

Bon voyage! ▲

OAG commands and sample displays reproduced with permission, courtesy Official Airline Guides, Inc., Oak Brook, Illinois 60521.

BOARDS & BUSES

by Kevin Goldstein

Though barely two years old, the pc marketplace showed its strength last month with not one but two shows devoted solely to the littlest IBM, one in New York and the other in San Francisco. The San Francisco show in particular turned out to be especially fertile hunting grounds for the wily hardware shopper.

Until IBM's introduction of its XT, consumers and manufacturers alike were delaying hard disk decisions. Manufacturers had no desire to spend thousands of dollars bringing a new disk product to market, only to discover later that it was incompatible with IBM's offering; consumers balked at parting with hard-earned cash for an expensive add-on that could turn out to be incompatible with the software IBM would release to support its own hard disk.

All that is history. With the XT and DOS 2.0 having been on the market long enough for their innards to have been thoroughly scrutinized by the makers of add-on hardware, the race is on to field a hard disk that gives the user the function of an XT at a fraction of the price.

10MB for under a Grand. If that's what you want, you would have hit pay dirt at PC '83 in San Francisco. Brazenly displayed over Great Lakes Computer Peripherals's booth was a sign that read, "IBM PC Compatible 10 Megabyte Hard Disk for under $1000."

That sign caused a lot of head-swiveling; up to now, the cheapest ten-megabyte disk has been in the range of $1,500 (with controller), and prices have been typically more like $2,000. So what's the catch?

The only apparent catch is that you'll be sending a lot of cash to a mail-order company—no chance to fondle the merchandise before you buy. The $1,000 price includes the drive with its own case and power supply, a hard disk controller card, plus an installable device driver that company president Richard Brown claims makes the hard disk completely compatible with DOS 2.0.

And here's the clincher: Great Lakes promises the drives will be genuine Shugarts. (For those of you not familiar with Shugart, let's just say that Al Shugart is an institution in the computer industry; Shugart Associates, which makes the drive Great Lakes will be selling, is to disk drives what IBM is to computers. Maybe a bit of an overstatement, but not much. Some people never stop. Al Shugart sold Shugart Associates and started Seagate; it too is now a very successful manufacturer of small disk drives.)

If all this is true, it means you could put together a "virtual XT" for well under the price IBM charges for its real XT. Let's assume you bought a 64K single-floppy-disk-drive pc for about $2,200. Now add 18 64K DRAMs (dynamic RAM memory chips) at maybe $6 each to bring motherboard memory up to the 128K total of the XT, plus an async board at maybe $100, plus $1,000 for the hard disk. Voila—for only $3,400 you'd have a system with all the power of an XT but at a cost of about $1,600 less.

And for another $500, Great Lakes says it will throw in a 32-megabyte tape backup drive.

That all sounds almost too good to be true. Those who've been using the XT come away really appreciating that hard disk. Not only does it end the floppy-disk swap game, it speeds up otherwise cumbersome compilations and assemblies by an incredible amount. It's thus with great anticipation that we look to reviewing Great Lakes's bargain. Stay tuned; this column will let you know if it's everything it's claimed to be. The test will focus on whether in use it appears to be an XT. Complete report to come.

Great Lakes is only six months old, so no claims can be made about its longevity, but if it can deliver what it promises, the firm should be in for a long and healthy life. For those of you who can't wait for the review, Great Lakes is in Hoffman Estates, Illinois.

By the bye, if you're a real memory hog, Great Lakes claims it will also be delivering disks of up to 140-megabyte capacity—that's on a standard 5¼-inch Winchester.

132 Columns You Can Read. Disks weren't the only news at the shows, however. One particularly interesting piece of hardware was a monochrome display board shown by California Computer Systems. And what can be unique about a monochrome display board, you wonder. Try display capabilities of 132 columns by 44 lines. Have you ever tried to decipher a 132-column assembler listing on an 80-column display? Users of large spreadsheets will be another appreciative market.

The most surprising thing about the display was not the number of characters it could display, however, but how very readable those tiny little characters turned out to

be. IBM's monochrome display never ceases to amaze.

Text on an RGB color monitor just doesn't compare to text on the monochrome screen. It's not that the characters put out by the graphics display adapter are that bad—it's that the characters generated by the monochrome board for the IBM display are that good. Part of the problem is in the display adapter, and the other in the display itself.

The problem with the display itself is fairly easy to understand: Color monitors with the kind of resolution necessary to approach the quality of IBM's monochrome display (over 350 horizontal lines) are very expensive. Thus the nine-dot-wide-by-fourteen-dot-high character matrix that the monochrome board generates is cut back to a mere eight by eight for the color adapter. Actually, that's not entirely accurate: Because the display adapter can generate 640 dots in the horizontal direction (from left to right on the screen), each horizontal "dot" is actually two identical dots. The matrix is thus really sixteen by eight, with the sixteen dots consisting of eight pairs of two. The board was set up this way because television and low-resolution composite displays can't resolve 640 horizontal pixels; that many pixels would simply smear together. But high-quality RGB displays can, and this double-dot approach means that the characters are not as sharp as they would be if a true eight-by-eight matrix were used.

The color graphics capabilities are both one of the best features and one of the major disappointments of the pc. Disappointments, because the vertical resolution in any graphics mode just isn't that good and because the choice of colors at even medium resolution is very limited. But color graphics are still a strong feature of the pc, because even within those limits a lot of good work can be done.

The reason for these limits is understandable; as we said, IBM designed the color board to be used with an affordable display. It's likely that as high-quality monitors come down in price, and as IBM sees that its users are willing to spend whatever bucks are required for high-resolution displays, the company may offer a higher-performance display adapter.

In the meantime, if you've got a worthy display, you can purchase such adapters from independent vendors like the Plantronics division of Frederick Electronics. Their display adapter boasts greater color selection at high resolution, and we'll be taking a closer look at it and similar boards in a subsequent column. Many of the boards offer other special features; the Plantronics board can generate single-dot-wide characters in graphics-text mode, and the Hercules and California Computer Systems boards can generate graphics on the monochrome monitor.

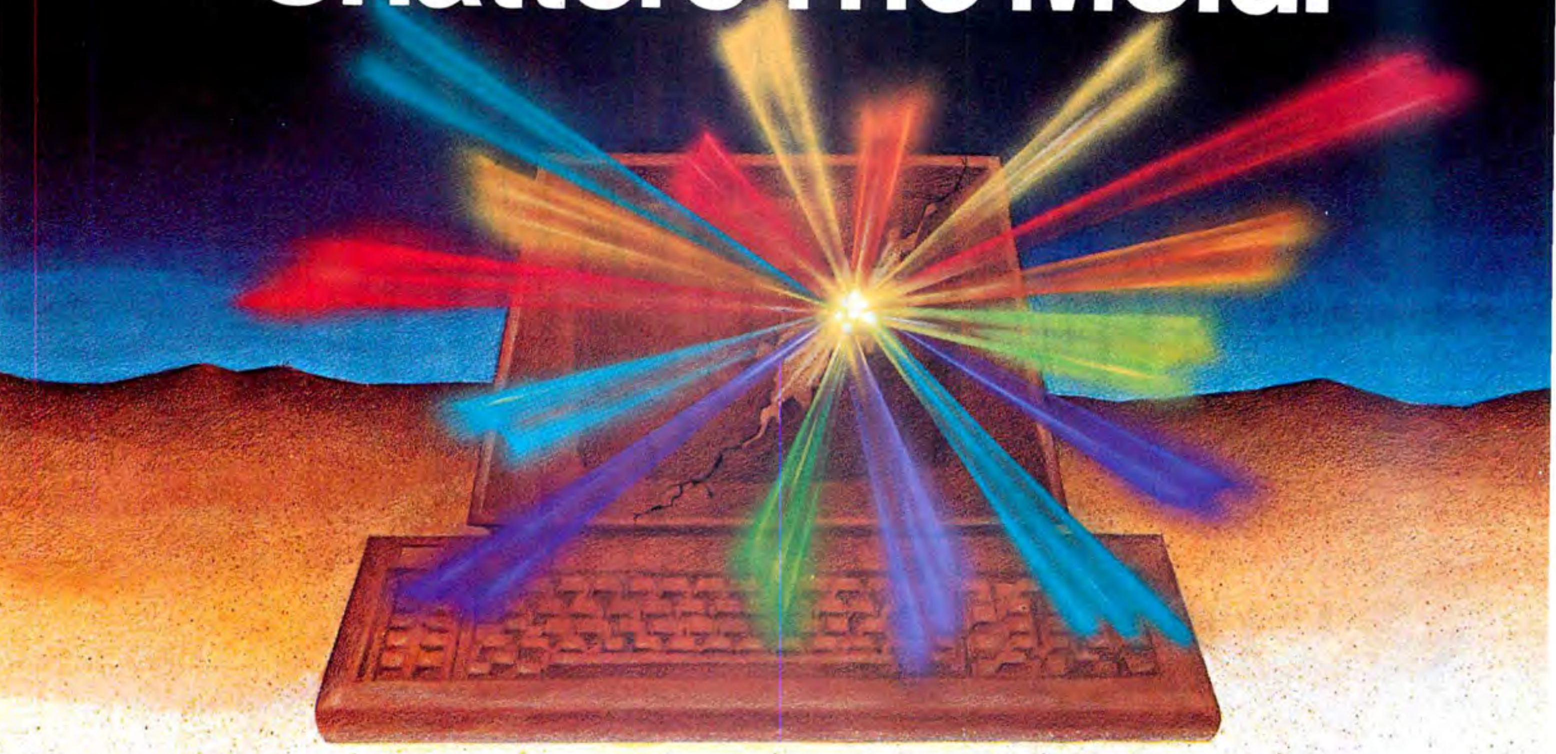
Speaking of color monitors dropping in price, which is a normal expectation for any product that becomes a volume-oriented consumer commodity, Amdek recently lowered the price of its Color II RGB-input monitor from $649 to $529. (The II offers 560-horizontal-by-240-vertical resolution.) Probably feeling the heat from IBM's display.

What with all the goodies available for your pc, it's likely sooner or later you're going to run out of expansion slots. Expansion chassis are now available from several companies; Tecmar was probably the first to offer such a chassis. If you don't need room for a hard disk and think six expansion slots will be enough, I-Bus of San Diego, California, has a six-slot expansion chassis with power supply for only $700. It's called—what else?—the Six Pac.

Rumor Has It . . . that as soon as IBM has worked off its current stock of disk drives, it will start supplying pcs with half heights. Let's see, that means two floppies and the hard disk could fit in the system unit of an XT, or up to four floppies in a standard pc. Not bad. Especially if Big Blue decides to use quad-density floppies. That may not be in the cards for the near future, however; the company still hasn't seen fit to run the drives with ten sectors per disk, which the controller and drive can both support. But if (or when) they do, you could pack over 3¼ megabytes of floppy disk storage into one pc system unit.

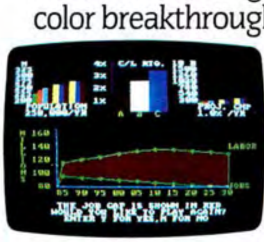
Micro-Meltdown, Part II. As noted in last month's column, heat is the primary enemy of solid-state components. As chip densities increase and more heat is generated in a smaller package, manufacturers will increasingly be moving to the technology known as CMOS. If you've poked around much in computer circuits, you know that five volts is the almost universal operating standard. That operating voltage is likely to shrink along with chip geometries; running a chip at a lower voltage saves power, and thus it generates less heat. Historically, and particularly with CMOS, there's been a tradeoff between maximum operating speed and power used: The faster you wanted to run a chip, the more power you had to apply. As technology improves, it becomes possible to squeeze the same speed out of less power; we're now pretty close to the point when it's likely that the ubiquitous five volts will drop to three or so. A side advantage to such a change is that such low-voltage circuits are easier to run on battery supplies; batteries generally put out 1.5 volts or so, and two of them would do nicely for a three-volt circuit. ▲

COLORPLUS™
Shatters The Mold.



A vibrant, varied new world of color is now at your fingertips.

We've shattered the mold for color graphics adapters with COLORPLUS, a high resolution, full color breakthrough fully compatible with IBM Personal Computer hardware and software.

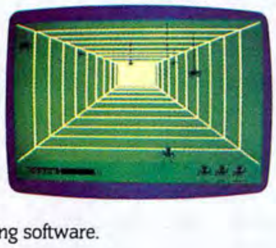


COLORPLUS produces incredibly sharp, detailed multi-color graphics of professional quality at a low PC price.* Its state of the art biplanar technology enables high resolution 4 color 80 character graphics or medium resolution 16 color 40 character graphics.

The current IBM PC capabilities of high resolution 1 color 80 character graphics and medium resolution 4 color 40 character graphics simply pale in comparison, now that the era of COLORPLUS has arrived.

XT COMPATIBLE

Using only a single expansion slot, COLORPLUS includes an integrated parallel port, thereby saving you an expansion slot for future needs. No modification is required to existing software in the IBM modes. Enhanced software supporting COLORPLUS Graphics capabilities is currently available from major software suppliers.



For over 31 years, Frederick Electronics Corporation has developed high quality electronic equipment world-wide. The key to our success is our international reputation for customer support. PC+ Products will continue in this tradition. Frederick Electronics Corporation is a subsidiary of Plantronics, Inc., a publicly owned company listed on the NYSE as PLX.

The goal of PC+ Products is to expand your options, to make your personal computer more valuable and productive. COLORPLUS is just one example of our commitment to that objective.

Contact us for the name of your nearest COLORPLUS dealer.

PC+™ PRODUCTS

PLANTRONICS™
Frederick Electronics

7630 Hayward Road, P.O. Box 502
Frederick, Maryland 21701
Telephone: 1-800-638-6211
(301) 662-5901, TELEX: 893438

* Suggested retail price: $475.00, including software.

PC+ and COLORPLUS are trademarks of Plantronics/Frederick Electronics Corp. IBM is a registered trademark of International Business Machines Corp.

YOUR IBM® PERSONAL COMPUTER IS ABOUT TO BECOME MORE POWERFUL THAN YOU EVER DREAMED POSSIBLE.

PC '83

BOSTON

Tuesday-Thursday, October 4-6, 1983
Bayside Exposition Center
10:30 AM to 5:30 PM Daily

Produced by Northeast Expositions Inc., the people who brought you Applefest, CP/M '83, Softcon, and the National Computer Shows.
826 Boylston Street, Chestnut Hill, Massachusetts 02167, 617-739-2000 or 800-841-7000.
For information on exhibiting at PC '83, please call 800-343-2222.

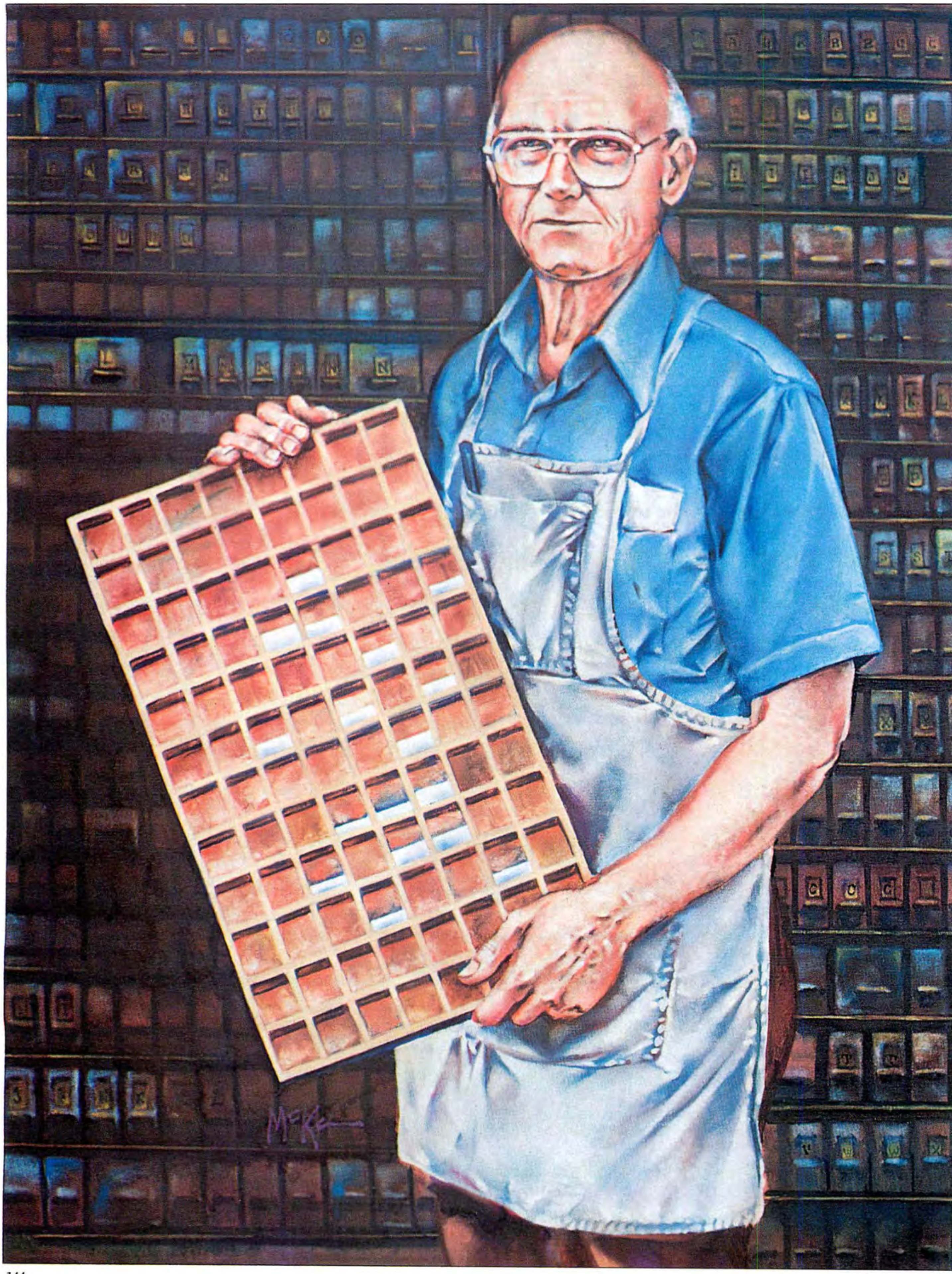


Illustration by Kevin McKeon

COLUMN DEBUT

John Dickinson has been involved with mainframes and minis for over 15 years and currently works with mainframes and micros at Manufacturers Hanover Trust.

The Printed Word

by John Dickinson

This column is the first in a monthly series dedicated to helping you get the most out of your IBM Personal Computer's printer. This may appear to be a simple subject, but it isn't. From the moment you decide to attach a printer to your pc, you're beset with complex questions to answer and things to learn.

What's so complex?

There are so many printers that will attach successfully to your pc that even if the only decision you had to make was the selection of the right printer for your applications, that decision might require a significant amount of research and time. After you buy your printer, using its full capabilities in the pc environment presents a new set of problems. No two printers have the same features, and no two printers are completely compatible with each other. The printer you buy may not be completely compatible with the software you use. Even buying the right kind of paper for your printer and applications can present you with difficult choices.

Our column will start out by giving you some background on printers and their use in the computer environment, and on how IBM and the pc fit into the printer environment. We'll also define some of the items and issues to be discussed in coming months. The subject of pc printers is very broad, so this introductory installment is necessarily general. Future columns will focus on specific issues in the area of pcs and printed words.

A Printer for Every pc. Just about every computer in use today has a printer attached to it. Your IBM pc probably is no exception. If you are a typical pc user, you have at least one printer hooked up, or you're thinking about getting one soon.

Most modern computer applications are geared to create interaction between the user (you) and the computer (your pc); the primary instruments of this interaction are a video screen and a keyboard. The keyboard-and-CRT combination makes a very responsive work environment and costs less than older printer-oriented interactive and batch systems, which used up a lot of paper and time. Most of today's exciting technological developments are in color video display and telecommunication of computer output.

Words printed on paper nevertheless still represent a substantial portion of the useful output produced by computer applications of all types. This may seem like an archaic state

of affairs, but the reason for it lies in the practicality and universal acceptability of printed words for communication.

Printed words are at least as important for communications today as in the past. You are probably reading this magazine to keep up with the IBM pc world. You undoubtedly read other magazines, newspapers, and books. One presumes that you see an occasional billboard or read a package label. All of these achieve communication by means of printed words.

Your pc's printer provides you with a method to output data as words and numbers printed on paper. No matter how large your output is, it can be produced in a form that others can easily view and understand wherever they may be—across the room or across the world.

Where do IBM and the pc fit into the world of printed words? IBM produces office typewriters and mainframe computer printers that have set industry standards for years. Relative to the rest of the company's printing equipment, their pc printer situation cannot be called a strong one.

When IBM entered the microcomputer business, it chose to use a processor (the Intel 8088) with sixteen-bit logic and an eight-bit data path. The reason for choosing an eight-bit data path was to make the pc adaptable to the wide variety of readily available microcomputer peripheral devices. One of those devices is the eight-bit microcomputer printer.

IBM selected the Epson MX-80 as the basis for the original pc Matrix Printer, which it later upgraded to the pc Graphics Printer by adding the Graftrax-Plus option with its own graphics character set and logic. To satisfy the need for a letter-quality printer, IBM arranged to become a distributor for NEC's 3550 Spinwriter Printer.

The Printer According to IBM. While there are plenty of rumors about a new pc printer from IBM, the situation is unchanged as of this writing. As a result, IBM has not provided pc users with a very strong definition of a pc-compatible printer. This definition will do for now: Any general-purpose eight-bit microcomputer printer with a parallel interface is an IBM pc-compatible printer.

What does that mean? The answer will be furnished in greater detail in later columns, but it generally means that almost any eight-bit printer available will work at some level. The rest of this month's column will demonstrate the complexity introduced by this definition.

The microcomputer industry is producing an awesome variety of eight-bit microcomputer printers with parallel interfaces that can be attached to your pc. There are about a hundred that will work, and new models are arriving at a rapid pace.

The fact that these printers work, however, is of limited significance. It means that, if you attach the printer to the proper connector on your pc by means of the proper cable, the printer will print your computer's character output. You will be able to print any character in the printable range of the ASCII character set. To tighten things up a bit, then, add the following to our definition of a pc-compatible printer:

A general-purpose eight-bit microcomputer printer is one that correctly produces the printable ASCII character set. The printable range in ASCII starts at character 32 (a space) and ends at character 126 (the tilde) (see "Beginners' Corner" starting in the May 1983 issue of *Softalk* for a full explanation of the ASCII character set).

The two basic types of general-purpose eight-bit microcomputer printer are dot-matrix printers and fully-formed-character printers. For each type there is a broad range of available technologies and capabilities.

Later columns will discuss some of the details, but the following two definitions will suffice for now: A dot-matrix printer is one that composes each character from a series of small dots printed on the paper; this procedure is similar to the way in which characters are formed on a video screen. A fully-formed-character printer is one that impacts the paper by means of a die cast in the shape of the character to be printed. The dies for each character are contained on a wheel, ball, drum, or thimble device, similar to those used on most modern office typewriters. The fully-formed-character printers are generally referred to as "letter-quality" printers. Although dot-matrix printers are not usually considered capable of producing high-quality output, there are some relatively new matrix printers that blur the traditional distinction by producing so-called "correspondence-quality" output.

Dot-matrix printers are generally faster than fully-formed-character printers; they're also usually quieter. These may be important considerations when you're choosing a printer for your work environment. They are by no means the only considerations, however, as we'll see in subsequent columns.

Within each category—dot-matrix and fully-formed-character—printers vary widely in capability. There are different mixes of features in each available printer, and the quality and usability of the features differ from one machine to the next.

Feature vs. Capability. You might reasonably wonder what's

meant by the terms *printer feature* and *printer capability*. The fact that there are so many different things called printer features makes the concept itself a little awkward to define. Part of the definition will have to come from you. How you intend to use your printer will help you decide whether a capability is truly a "feature" (that is, something that enhances the utility of the printer) or just something that costs extra but serves no purpose in your printer applications.

Printer features may affect a number of different printer functions, including character image (can you control the darkness of the print, can you underline or produce characters in varying widths and fonts?), vertical or horizontal spacing and tabulation, form (page) size, and more general facilities such as ribbon color, print speed, and paper-feed modes. Printer features may also include the ability to dump monochrome or color graphics; and some printers allow the mixing of character and graphics output.

Deciding which features are right for your applications may be difficult. It's hard enough just to understand exactly which features are available on a particular printer. The marketing literature that exists for printers usually describes their more exotic features, such as graphics, and ignores the more mundane (and very useful) ones, such as character underlining and forms control. IBM is a case in point. The current IBM literature describes IBM's original model and ignores completely the significant enhancements available on its graphics printer.

Sales personnel are not very knowledgeable either, but it's not their fault. They too can learn about printer features only from the marketing literature available from the manufacturers.

Using the features of your printer is an even thornier problem. There are three reasons for this:

(1) There is nothing approaching compatibility in printer feature design. A printer feature may be implemented in any way chosen by its designers, and designers all have their own ideas of what a feature ought to do and how it should be used. The compatibility problem sometimes exists even within a single manufacturer's product line. For example, IBM's original matrix printer and its current graphics printer are incompatible in some areas. The same can be said about Epson's MX-80 series and the newer FX-80 and RX-80 printers. None of the Epson printers is fully compatible with the IBM versions.

(2) Documentation of printer features is almost universally hard to understand. Printer documentation seems to be written for engineers and hobbyists—guys who really care which pins carry which signals between the computer and the printer.

For the pc user whose only interest is word processing or spreadsheets, though, there is little or no help in the documentation supplied with the printers. Even the most technically adept users may have to guess what a feature actually does and how to use it. Almost all printer documentation assumes that the user knows how to program in Basic (and uses nothing else) and is well versed in ASCII codes and "escape sequences."

(3) Documentation on how to use your printer's features in conjunction with software products and languages can be almost as obscure and difficult to use. There are some notably well documented printer output interfaces, but these are exceptions.

Software product documentation often ignores printer output (beyond a simple discussion of how to print a report). Printer features are either not supported or are supported for a printer you don't have. Some software products support only a limited range of printer models and have to be patched before they'll support yours. Almost all IBM-supplied software documentation assumes that you are using IBM's printer (or a "compatible" one).

Many software products simulate printer features, such as overstrike, underscore, and form feed, that were not available on earlier microcomputer printers. Many of today's printers have these features, and the decision whether to access the feature through software or directly through the printer may be a difficult one. Some software simulations produce better output than certain printers' hardware features, so experimentation is required. With some software products, only the software simulation of printer features is available, so no decision is required.

The IBM Personal Computer printer situation can be summarized in the following way:

A wide variety of printer types and models are available for the pc.

The printers are all somewhat, but not completely, compatible.

Information describing printer features is difficult to obtain.

There is a great deal of uncertainty about IBM's next move in pc printers.

Documentation of many printers is weak, to say the least.

Software interfaces to printers are not well defined or documented.

There are many topics to take up in this column, and plenty for you to learn. We'll get into issues ranging from the selection of the right type of printer for various applications to how to use printers, how they work, and what your printer documentation really says. Some good (and bad) examples of software and language documentation will be described along with how to use your printer in those environments. Paper products, ribbons, and printer maintenance are also on the schedule for discussion. If a particularly exciting new product should come out, it will probably be reviewed here.

Next month's installment will discuss how your pc communicates with your printer, including a nontechnical explanation of the standard parallel interface. We'll look at how the eight-bit ASCII code is used both for printing characters and controlling your printer, and we'll include a clear explanation of escape sequences. ▲

THE ONE AND ONLY

CRAMBO

You've got a problem. Go to your favorite computer store. Ask to see their Combo Cards. Then ask about their RAM Cards. See the problem? Just too many to pick from. Now, ask to see the CRAMBO™. No problem. There's only one. Ours. Except you don't know what a CRAMBO is. You've probably guessed it's a Combo Card piggy backed to a RAM Card. So it only takes up one slot in your IBM/PC. That's right. But here are the details.

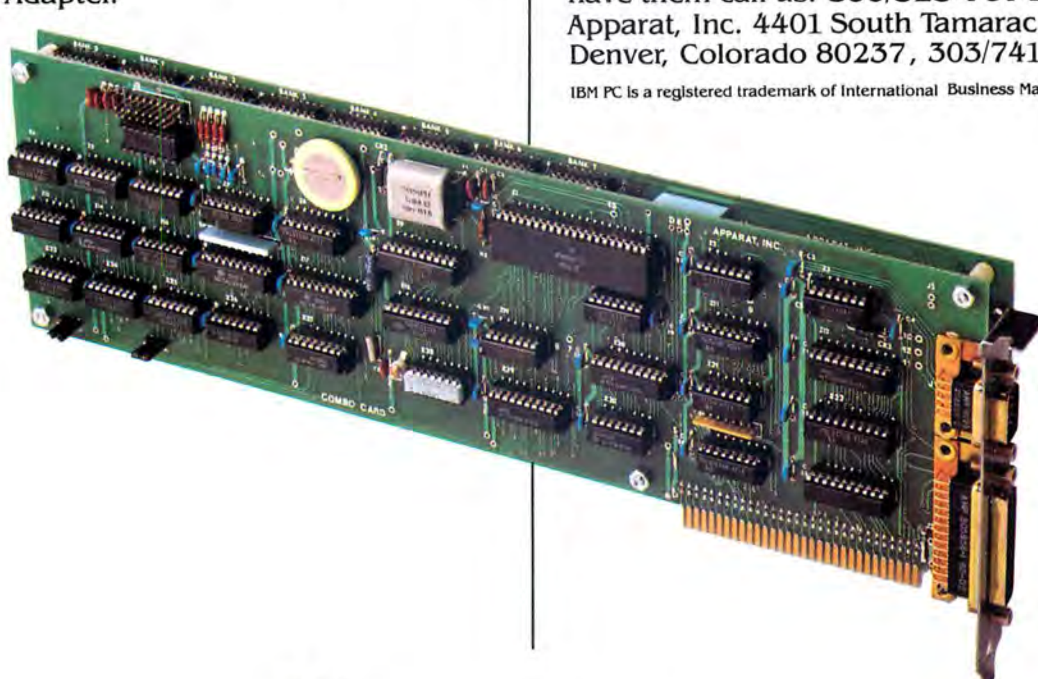
THE COMBO II CARD:
For **$189** you'll get a Clock Calendar, Async Communications, Parallel Printer and a Game Adapter.

THE 512K RAM CARD:
It too, is only **$189** with 64K of RAM installed. And when you need more RAM, 64K increments are available for $64 each. And, SDRIVE, the electronic disk emulator, is available at no cost with the RAM Card.

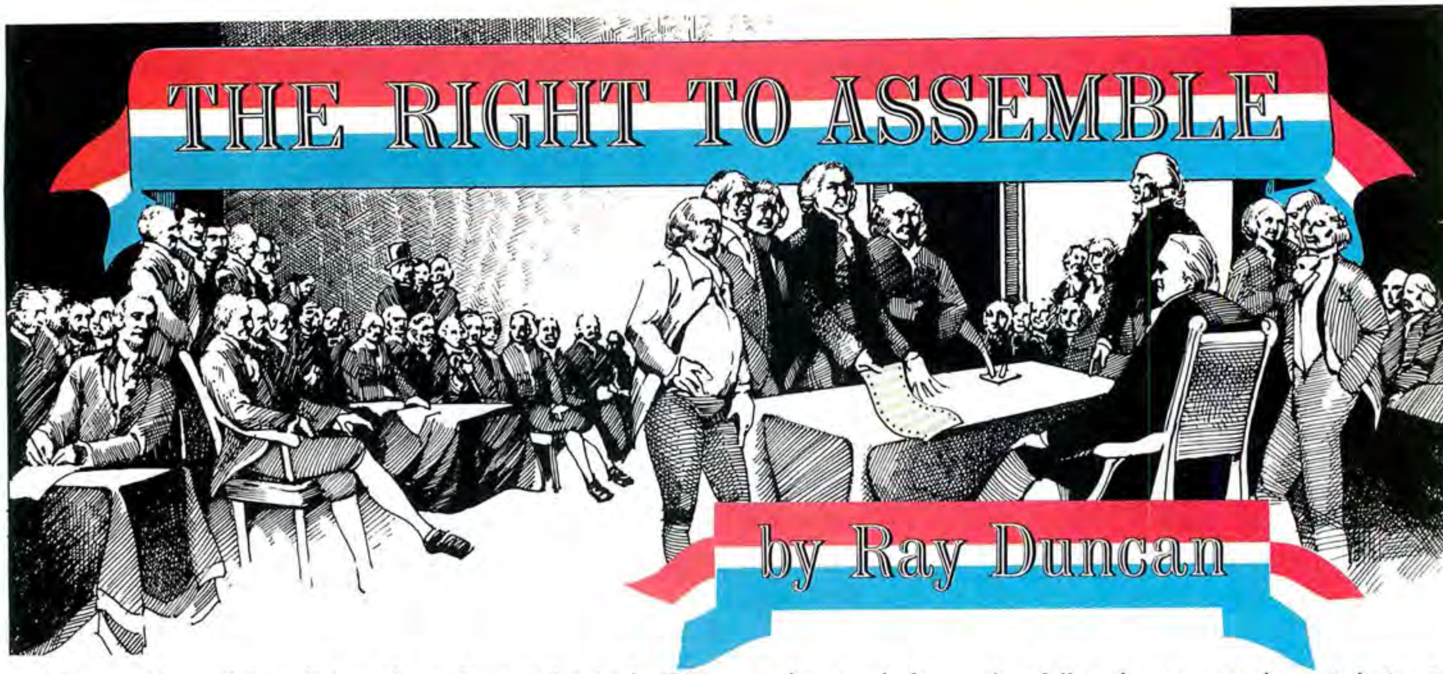
You can buy either of our boards separately for use in the XT or PC expansion chassis. But only our boards can be piggy backed to give you the one and only CRAMBO. And for only **$359.**

Go check out the CRAMBO at the same store carrying all those Combo Cards and RAM Cards. If they don't have the CRAMBO have them call us. **800/525-7674.** Or write: Apparat, Inc. 4401 South Tamarac Parkway, Denver, Colorado 80237, 303/741-1778.

IBM PC is a registered trademark of International Business Machine Corp.



Apparat, Inc.



THE RIGHT TO ASSEMBLE

by Ray Duncan

This month we'll launch into the real business of assembly language programming and lead you step by step through the creation, assembly, linking, and execution of a short utility. The program presented is called Memsize and was contributed by Rick Wilton. It will be useful to those readers with lots of RAM; and it will illustrate the use of the Microsoft assembler.

Memsize was written because the sense switches in the IBM pc can be set only for a maximum of 544K of RAM, even though the published memory map allows the user to add RAM up to the address A0000H (656K). Fortunately, the operating system reads the sense switches only once, at bootup, and it stores the RAM size in a dedicated memory location that it refers to thereafter.

When Memsize is invoked, it first checks the memory size variable to see if it is set for less than 544K. If that's the case, the variable may have been modified by an electronic disk program to hide some of the RAM from the operating system. To avoid causing problems in this instance, Memsize simply exits. If the memory size variable is indeed set for the maximum, 544K, the Memsize program physically sizes memory by attempting a write/read cycle on each sixteen-byte boundary up to the A0000H limit. When the address of the true top of memory is found, it is shifted right to divide it down to kilobytes, then stored into the operating system's reserved location. The most logical way to use Memsize is to add it to your Autoexec.bat file, where it will automatically be executed on every system startup.

Listing 1 gives the source code for Memsize. You should use your favorite screen or line editor to enter this, *exactly* as you see it printed, into a file named Memsize.asm. If you're using *WordStar*, be sure to create the file in nondocument mode. After you have entered the source code, proofread it carefully, then transfer it to a disk in drive B. You will also need the following other files on the same disk: Asm.exe, Cref.exe, Exe2Bin.exe, and Link.exe.

Now log on to drive B (if you haven't already) by entering:

```
A>B;
B>
```

then follow the sequence shown in listing 2. The underscored words in that listing are for you to enter; the rest is responses from the assembler. Every message from the assembler should print out exactly as you see it here! If you get any warning errors or severe errors from the assembler, you probably made some typing errors when you were entering the source code; go back and check it against listing 1 again. Note that the assembler will not give you any warning at all if you omitted entire lines or entered them out of order; to detect this sort of error you must simply compare your program listing with what's shown here.

The second step shown in listing 2 is to run the Cref (cross reference) utility program, which creates a listing that can be very useful when you're debugging large assembly language programs.

The third step in listing 2 is *linking*. This converts the file output by the assembler into a form that can be loaded by the operating system; and the result is a file named Memsize.exe. When you run the linker this time, you're going to get an error message warning you of a missing stack segment; you can ignore it.

The final step in listing 2 is to invoke the Exe2Bin utility to convert the Memsize.exe file into a Memsize.bin file, which can be renamed to a com file. Com files are more compact and do not require relocation by the operating system at load time.

At the end of listing 2, you'll find a directory of the various Memsize files that were created during the process described above.

Memsize.asm is the original source file, created by you!

Memsize.lst is the program listing generated by the assembler; it includes all of your original source program, the machine code generated, line numbers, and any error messages.

Let Babytalk™ do the talking.

Babytalk is an intelligent communications management interface packed with features. In fact, it's a 64K Z-80 based microcomputer that plugs into your IBM PC to manage a full range of communications:

AS A TERMINAL EMULATOR,
Babytalk emulates 3270/bisync and 3270/SNA, 2770, 2780, 3770, 3780 BATCH/BISYNC, and HASP/BISYNC and a wide range of ASYNC terminals including VT-52, VT-100, IBM 3101, and IBM 5251. It's KEYFIX utility automates your keyboard by defining function keys with any multi-keystroke sequence up to 80 characters.

AS A SUPERSMART MODEM,
Babytalk's onboard 300/600/1200 Baud auto-dial, auto-answer, direct-connect MODEM supports Bell 103, 202 and CCITT V.21 and V.23 protocol.

AS A SPOOLER,
Babytalk's serial or parallel port may be used for background print spooling, so that your IBM PC can run applications while Babytalk manages print output to either a serial or parallel printer. Babytalk also has a Realtime Clock/Calendar with battery backup for programmable message dispatching, and so that you don't have to set time/date each time your system boots.

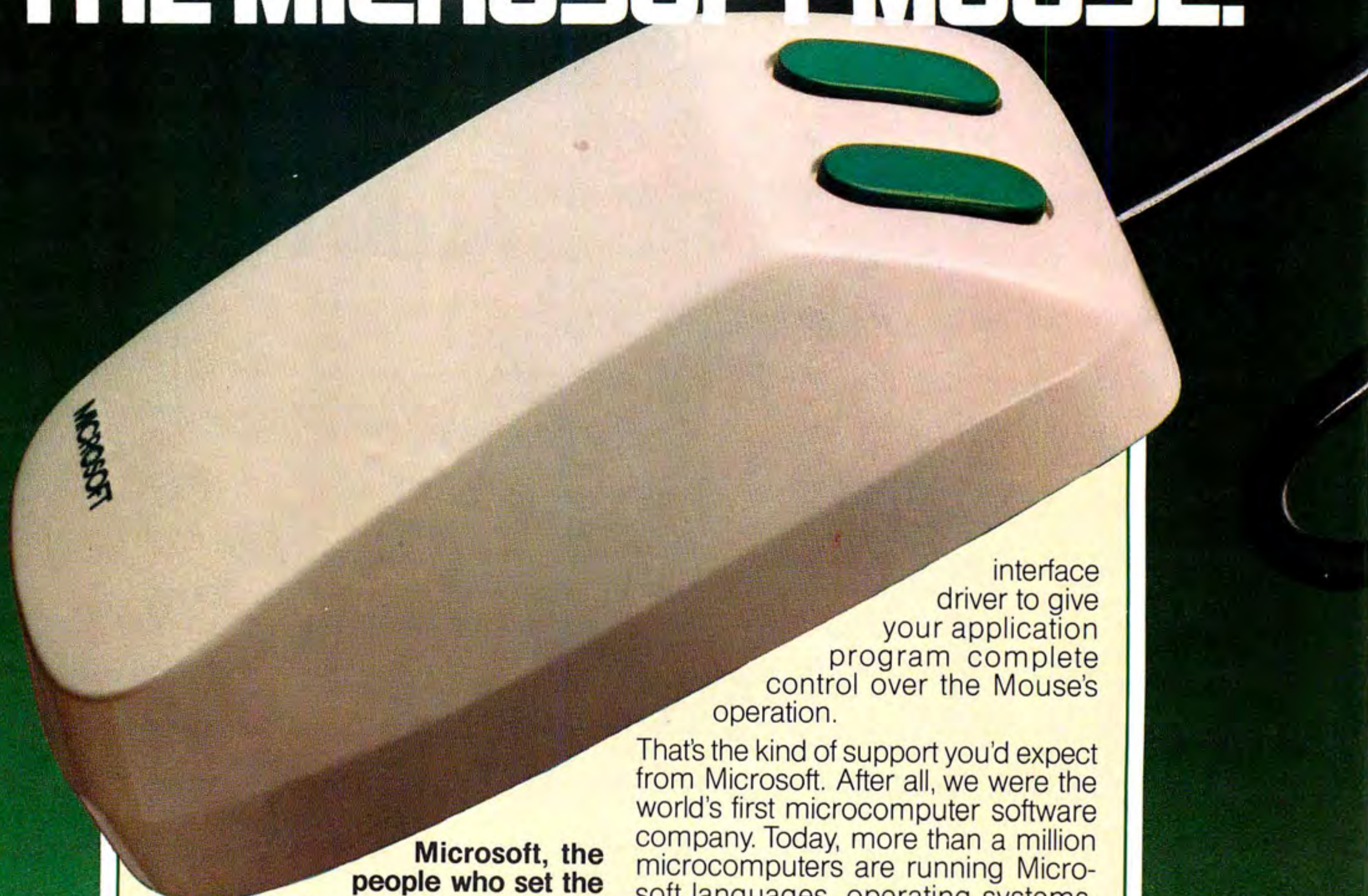
Babytalk is from the same people who gave you Baby Blue. It's available for the IBM PC, the Texas Instruments Professional Computer, and other MS-DOS compatibles.

For more details see your computer dealer or contact Microlog at (914) 368-0353, Microlog, Inc., 222 Route 59, Suffern, NY 10901.

Only $895.

Microlog
A TU INTERNATIONAL COMPANY

THE NEW STANDARD.
THE MICROSOFT MOUSE.



interface driver to give your application program complete control over the Mouse's operation.

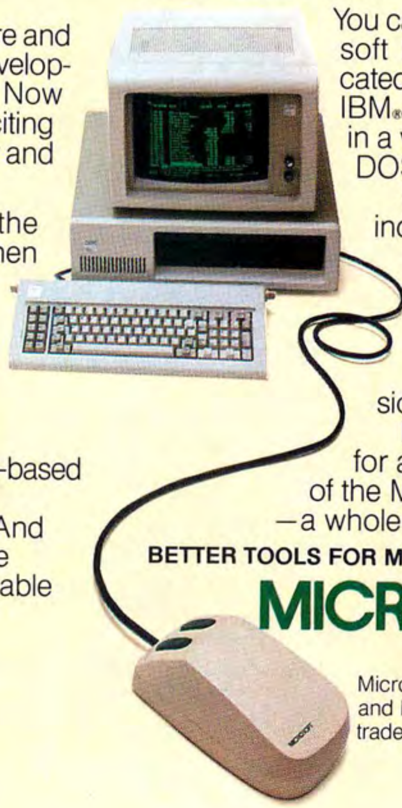
That's the kind of support you'd expect from Microsoft. After all, we were the world's first microcomputer software company. Today, more than a million microcomputers are running Microsoft languages, operating systems, application programs, and hardware-software combinations.

Microsoft, the people who set the standard for software, have done it again with the Microsoft® Mouse.

Our expertise in both hardware and software has gone into the development of the Microsoft Mouse. Now you can plug in the most exciting computer product of the year and put it to work.

The Mouse lets you move the cursor freely and naturally, then execute commands at the push of a button.

The Microsoft Mouse is a complete system. It comes with an on-screen tutorial, a practice application, and the Multi-Tool™ Notepad, a mouse-based text editor, so you can begin using the Mouse right away. And for application developers, the Mouse includes a programmable



You can get the Microsoft Mouse in dedicated versions for the IBM®-PC, PC XT, and in a version for MS™-DOS machines with serial interfaces, including the IBM-PC. The Mouse supports all versions of MS-DOS, including version 2.0. Ask your Microsoft dealer for a demonstration of the Microsoft Mouse —a whole new standard.

BETTER TOOLS FOR MICROCOMPUTERS

MICROSOFT™

```
        name    memsize
        page    55,132
        title   'MEMSIZE — reset PC/DOS memory size variable
;
; MEMSIZE — reset PC/DOS memory size variable at location 40:13h
; ( for IBM PC under PC-DOS 1.1 — See Technical Reference p. A-2 )
; Version 1     April 1983
;
; Richard Wilton
;

cseg    segment byte
        assume  cs:cseg,ds:cseg
        org     100h

start:  mov     bx,40h          ; set DS:BX to point ...
        mov     ds,bx
        mov     bx,13h          ; ... to memory size variable
        cmp     word ptr [bx],220h ; is current mem size < 544K ?
        jl      exit1           ; exit if it is

        mov     ax,8800h        ; otherwise, set up test loop
        mov     bx,0
loop1:  cmp     ax,0a000h       ; is AX = beginning of "reserved"addrs ?
        je      exit2           ; yes, so exit
        mov     ds,ax           ; no, so use this as a segment
        mov     [bx],ax         ; write contents of AX to DS:BX ...
        mov     cx,[bx]         ; ... and read it back to CX
        cmp     ax,cx           ; does data read = data written ?
        jne     exit2           ; exit if it doesn't
        mov     ax,ds           ; otherwise, copy DS to AX ...
        inc     ax              ; ... increment it ...
        jmp     loop1           ; ... and loop

exit2:  mov     cl,6            ; number of bits to shift
        shr     ax,cl           ; convert segment to Kbytes
        mov     bx,40h          ; again, set DS:BX ...
        mov     ds,bx           ; ... to point to memory size variable
        mov     bx,13h
        mov     [bx],ax         ; store updated memory size
exit1:  int     20h             ; exit to DOS
cseg    ends                    ; end of code segment

        end     start           ; end of assembly — start addr at "start"
```

List 1. Source size for Memsize.asm.

```
B>ASM MEMSIZE
The IBM Personal Computer Assembler
Version 1.00 ©Copyright IBM Corp 1981

Object filename [MEMSIZE.OBJ]:
Source listing  [NUL.LST]: MEMSIZE
Cross reference [NUL.CRF]: MEMSIZE

Warning Severe
Errors  Errors
0       0

B>CREF
Cref filename [.CRF]: MEMSIZE
List filename [MEMSIZE.REF]:

B>LINK MEMSIZE

IBM Personal Computer Linker
Version 2.00 ©Copyright IBM Corp 1981,1982,1983

Run File [MEMSIZE.EXE]:
List File [NUL.MAP]: MEMSIZE
Libraries [.LIB]:
Warning: No STACK segment

There was 1 error detected.

B>EXE2BIN MEMSIZE.EXE

B>DIR MEMSIZE.*

 Volume in drive B has no label
 Directory of B:\

MEMSIZE  OBJ     105   7-03-83  12:09a
MEMSIZE  LST    3377   7-03-83  12:09a
MEMSIZE  CRF     128   7-03-83  12:09a
MEMSIZE  REF     333   7-03-83  12:10a
MEMSIZE  MAP     256   7-03-83  12:10a
MEMSIZE  ASM    1936   7-03-83  12:04a
MEMSIZE  EXE     896   7-03-83  12:10a
MEMSIZE  BIN      56   7-03-83  12:10a
         8 File(s)  162816 bytes free
```

Listing 2. Audit trail of assembly and linking.

Memsize.crf is the raw data provided by the assembler for processing by the Cref utility.

Memsize.ref is the cross-reference listing generated by the Cref utility; it can be dumped to your terminal with the *type* command or to the printer with *copy*.

Memsize.obj is the machine-code output from the assembler, which is used by the linker to produce an executable file.

Memsize.exe is the file created by the linker, which can be loaded and executed as a transient program.

Memsize.map, a printable file produced by the linker, gives information about the segments, code groups, and entry points of your program.

```
The IBM Personal Computer Assembler 07-03-83        PAGE      1-1
'MEMSIZE — reset PC/DOS memory size variable


                            name    memsize
                            page    55,132
                            title   'MEMSIZE — reset PC/DOS memory size variable

                    ;
                    ; MEMSIZE — reset PC/DOS memory size variable at location 40:13h
                    ; ( for IBM PC under PC-DOS 1.1 — See Technical Reference p. A-2 )
                    ; Version 1     April 1983
                    ;
                    ; Richard Wilton
                    ;
                    ;
                    ;
                    ;

0000                cseg    segment byte
                            assume  cs:cseg,ds:cseg
0100                        org     100h

0100  BB 0040       start:  mov     bx,40h          ; set DS:BX to point ...
0103  8E DB                 mov     ds,bx
```

Memsize.bin is produced by the Exe2-Bin.exe utility, using the Memsize.exe file as input. It can be renamed to a com file.

You can spool the various printable listings to your printer with the following commands:

```
B>COPY MEMSIZE.ASM PRN:
B>COPY MEMSIZE.LST PRN:
B>COPY MEMSIZE.MAP PRN:
B>COPY MEMSIZE.REF PRN:
```

This is how listings 1, 3, 4, and 5 respectively were generated. You will be happier with the results if you set up your printer for 132-character-wide mode before attempting to print Memsize. lst.

Let's take a closer look at listing 3, which was produced by the assembler as it processed your source program.

The very first line on the page gives the name and date of creation of the assembler version you are using, along with the listing page number. The next line is called the title line, and contains whatever text you specify with the TITLE pseudo-op. From here on, we see actual program source code with some new, strange-looking columns of numbers appended.

The leftmost column consists simply of line numbers, which are assigned in sequence by the assembler as it reads in each line of your source code. These line numbers will be handy when you are attempting to look things up from the cross-reference listing, and also when you are trying to impress your manager with how much work you accomplished during the day. The second column of numbers contains the first memory address for each assembler instruction in your source code. These addresses are relative to the segment or group that the assembly language segment occurs in; they are not the same as the actual physical memory addresses where the machine code resides at execution time.

The third and fourth columns of numbers contain the actual machine code generated by the assembler from your source code. These are the binary instructions that tell the computer what to do; on the listing they are expressed in hexadecimal notation. Note that some instructions occupy only one byte, while others occupy as many as four bytes when a complex addressing mode or immediate data value is involved. In some cases (not in this specific program), when absolute address references are involved that must be resolved by the linker, the machine code you see on the listing will not be the same as the code that is finally loaded into memory and executed.

Next month we'll begin to discuss the various instruction groups of the 8088 microprocessor in detail, and we'll move on to even bigger and better programs! ▲

```
23  0105  BB  0013           mov   bx,13h          ; ... to memory size variable
24  0108  81  3F 0220        cmp   word ptr [bx],220h ; is current mem size < 544K ?
25  010C  7C  28             jl    exit1           ; exit if it is
26
27  010E  B8  8800           mov   ax,8800h        ; otherwise, set up test loop
28  0111  BB  0000           mov   bx,0
29  0114  3D  A000    loop1; cmp   ax,0a000h       ; is AX = beginning of "reserved" addrs ?
30  0117  74  0F             je    exit2           ; yes, so exit
31  0119  8E  D8             mov   ds,ax           ; no, so use this as a segment
32  011B  89  07             mov   [bx],ax         ; write contents of AX to DS:BX ...
33  011D  8B  0F             mov   cx,[bx]         ; ... and read it back to CX
34  011F  3B  C1             cmp   ax,cx           ; does data read = data written ?
35  0121  75  05             jne   exit2           ; exit if it doesn't
36  0123  8C  D8             mov   ax,ds           ; otherwise, copy DS to AX ...
37  0125  40                 inc   ax              ; ... increment it ...
38  0126  EB  EC             jmp   loop1           ; ... and loop
39
40  0128  B1  06      exit2: mov   cl,6            ; number of bits to shift
41  012A  D3  E8             shr   ax,cl           ; convert segment to Kbytes
42  012C  BB  0040           mov   bx,40h          ; again, set DS:BX ...
43  012F  8E  DB             mov   ds,bx           ; ... to point to memory size variable
44  0131  BB  0013           mov   bx,13h
45  0134  89  07             mov   [bx],ax         ; store updated memory size
46  0136  CD  20      exit1: int   20h             ; exit to DOS
47  0138              cseg   ends                  ; end of code segment
48
49                           end   start           ; end of assembly — start addr at "start"


          The IBM Personal Computer Assembler 07-03-83        PAGE        Symbols-1
          'MEMSIZE — reset PC/DOS memory size variable

          Segments and groups:

                  N a m e                     Size    align   combine class

          CSEG .............                  0138    BYTE    NONE

          Symbols:

                  N a m e                     Type    Value   Attr

          EXIT1 .............                 L NEAR  0136    CSEG
          EXIT2 .............                 L NEAR  0128    CSEG
          LOOP1 .............                 L NEAR  0114    CSEG
          START ............                  L NEAR  0100    CSEG

          Warning Severe
          Errors  Errors
          0       0
```

Listing 3. Program list for Memsize.

From "16-bit Toolkit," by Richard Wilton, appearing in the July 1983 issue of Dr. Dobb's Journal. *© 1983 People's Computer Company, Inc.*

```
Warning: No STACK segment

 Start      Stop      Length   Name             Class
 00000H     00137H    0138H    CSEG

 Origin   Group

 Program entry point at 0000:0100
```

Listing 4. Program map produced by the linker.

```
Symbol Cross Reference                    (# is definition)      Cref-1

CSEG .............            17#    18     18     47

EXIT1 .............           25     46#
EXIT2 .............           30     35     40#

LOOP1 .............           29#    38

START .............           21#    49
```

Listing 5. Cross-reference listing produced by Cref.

THE PROFIT PLOT
by Jack Grushcow

Last month we rolled up our sleeves and tackled the *Lookup* and *Choose* spreadsheet functions. This month we have a tutorial on logic functions.

There is a great deal of power in logic functions; they allow you to teach your spreadsheets to think. Using them, you can make decisions based on values elsewhere in your spreadsheet.

For instance, suppose you use a spreadsheet to forecast sales. Your costs are fixed up to a certain number of units produced. After that point, economy of scale reduces per-unit cost. You can use a logic function to check whether the forecast unit sales are over this threshold amount. If they are, you can apply the lower unit cost.

Calculating sales commission, discounts, tax rates, and many other business applications requires that your spreadsheet make a decision based on a result in a different part of your model.

The family of logic functions has four members: *If*, *And*, *Not*, and *Or*. Used together, they can give your spreadsheets a surprisingly high IQ.

The *If* function is the most frequently used. In its simplest form it makes an either/or decision based on the contents of certain spreadsheet cells.

If allows your spreadsheet to cope with many complex business events. For example: A salesman with more than $10,000 in sales earns a 3 percent commission; with sales below $10,000 he earns 2 percent.

You could solve this problem with the *Lookup* function, but your spreadsheet would soon be littered with lookup tables. Either/or decision-making is the strong suit of the *If* function. Because there are only two possible outcomes for this problem—the commission may be either 2 percent or 3 percent—*If* is the right tool for the job.

In its simplest form, *If* looks like this:

IF(CONDITION,ARG1,ARG2)

The *If* statement works in two distinct steps. First, it examines the condition found at the start of the expression. This condition can be a formula or expression that performs a test. For example, a condition such as A5>100 asks the question: Are the contents of cell A5 greater than 100?

Next, *If* selects arg1 or arg2 and displays the result. If the condition is true, *If* displays arg1; if false, arg2. That's where the either/or comes in.

To use *If* effectively, you must make the condition ask the right question.

Conditions, which can be as complex as necessary, have one feature in common: They all use one or more of the symbols <, >, or =. These symbols, which help create the condition we want to test, form the language of logic comparisons. They're often given foreboding-sounding names, such as Boolean functions or relational operators—a mouthful for just a few symbols.

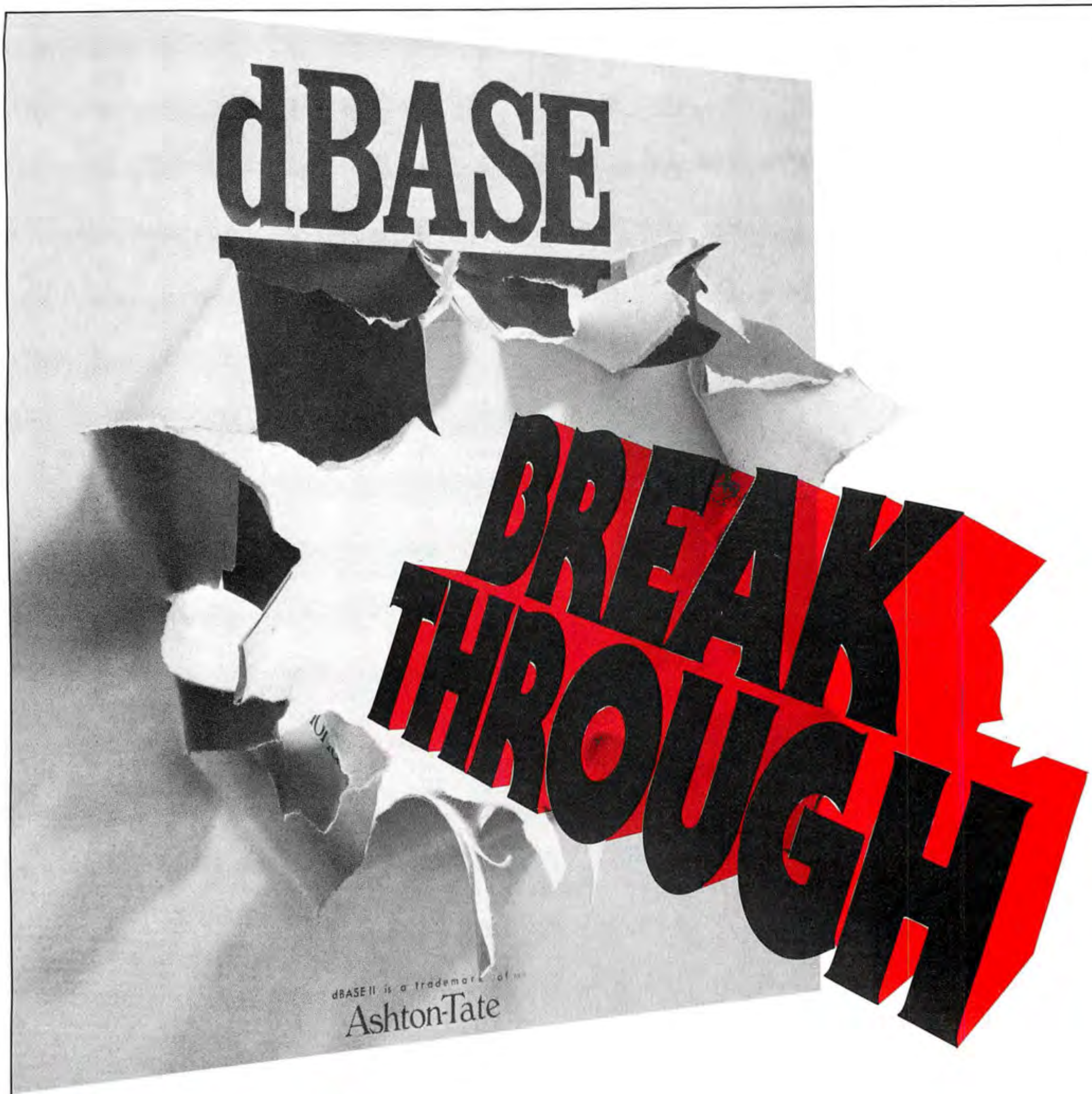
Placed between the left and right halves of a condition, these symbols ask one of several possible questions. Here is your spreadsheet's vocabulary of question-building blocks:

> Greater than?
< Less than?
= Equal to?
<= Less than or equal to?
>= Greater than or equal to?
<> Not equal to?

Using *If*, you can formulate thousands of different questions. The following examples could be found in the condition section of the *If* function:

SIN(B33)>=K22
LOOKUP(A5,B10...B20)*105<2340
SUM(A1...A25)+1000<>INT(K22)*SUM(K1...K12)

Let's look at an example of the *If* function and how it works, using this spreadsheet:


dBASE
BREAK THROUGH
dBASE II is a trademark of
Ashton-Tate

	A	B
1	Sales	:
2	Commission%	: IF(B1)>10000,.03,.02
3	Amount due	: +B1*B2

This example illustrates how to calculate a salesperson's commission by means of the *If* function. A commission of 3 percent of sales is paid if the monthly sales exceed $10,000. If they do not, the commission is 2 percent.

The *If* formula at B2 first examines the contents of cell B1. If B1 holds a value greater than 10000 (the sales are over $10,000), the condition is true and the value .03 is displayed in cell B2. If sales are less than or equal to $10,000, the condition is false and the value .02 is displayed in cell B2. The result of the *If* statement, whether 2 percent or 3 percent, is then used to calculate the commission owed.

We've had a progression in our analysis of spreadsheet functions, which began last issue. Let's take a step back and put this progression into perspective.

The *Lookup* function, which scans a lookup table to select a value from a list of values, selects from a list. It is useful when you have several possible choices with wide-ranging values. *Lookup* is a good way to select a tax rate from a table.

Choose also picks from a list, but it uses an index to select a specific value to display. It is useful when your selection is based on a specific outcome. Selecting an amount from a specific row or column is a common use. For example, *Choose* can select a unit sales figure from a specified quarter.

If narrows the scope of selection. It picks from a list of only two possible arguments, selecting one or the other depending on whether the condition is true or false.

If is useful in a spreadsheet full of formulas that display ERROR if data has not been entered. Dividing by zero is the common cause of such an error-message display; you can't calculate a percentage until you have some data to work with. New users are often dismayed to load a spreadsheet sprinkled with the message ERROR before they have even touched a key.

You can use *If* in combination with the *@Iserr* function to get rid of this daunting display. *@Iserr* returns a true value if the formula is evaluated to an error; it will, when used with *If*, display a friendly zero instead of the error.

This *1-2-3* spreadsheet shows how (*VisiCalc* users will use *@Iserror* rather than *@Iserr*):

	A	B	C
1		$	
2	Sales		@IF(@ISERR(A2/A2)*100,0,(A2/A2)*100
3	Cost of Sales		@IF(@ISERR(A3/A2)*100,0,(A3/A2)*100

The other three logic functions—*Not*, *And*, and *Or*—are often used with *If* to make complex decisions.

In *1-2-3*, the *And* function takes two arguments (*VisiCalc's And* takes as many as you like). It returns a true value if all the conditions are true, and false if any of the conditions are false. Its general form is:

AND(cond1,cond2)

For *VisiCalc*, it's:

@AND(cond1,cond2,cond3,...)

The *Or* function is similar to *And*. A value of true is returned if any condition is true, and a value of false if all the conditions are false. Its form is the same as that for *And*; just substitute *Or* for *And*.

The final function we'll discuss is *Not*. Its general form is:

NOT(Condition)

Not will flip the condition's value to true if it is false, or from false if it is true.

Not, *And*, and *Or* most often are used in conjunction with *If*. Let's look at some specific examples.

If fourth-quarter sales are less than 500 units, the ad budget will

	M	N	O	P
1	Month:	Oct	Nov	Dec
2	Sales			
3	Ad Expense	6000	6000	@IF(@AND(N1>500,O1' >500, P1>500) 6000,10000

Figure 1

be increased from $6,000 to $10,000. Figure 1 (*VisiCalc*) uses *And* to create an expanded condition. This condition is true only if each condition in the *And* list is true.

We analyze logic formulas the way we examine numerical formulas: we peel away the layers of brackets and work our way out from the inside. In the case of the formula stored at P3, we first *And* the three conditions: N1>500,O1>500,P1> 500.

And will return a true value only if all the conditions are true. In our problem, there must be more than 500 units sold in each of the last three months; a value of false will be returned if sales for any of the months drop below 500. After peeling away the most interior set of brackets and performing the *And*, we can think of the formula as:

@IF(TRUE,6000,10000)

Now we have a simpler looking formula. Since sales are over 500 units for each of the last three months, the condition is true and the value 6000 is selected. Should sales in any one of the three months fall below 500 units the *And* condition will be false and the value 10000 will be selected.

Lotus's *1-2-3* uses a slightly different approach to logic functions. They are inserted directly into the formulas and the # (octothrope) is used to denote each logical operator. Figure 2 shows the *VisiCalc* spreadsheet in figure 1 redone for *1-2-3*.

	M	N	O	P
1	Month:	Oct	Nov	Dec
2	Sales			
3	Ad Expense	6000	6000	@IF(N1>500#AND#O1 >500#AND#P1>500), 6000,10000

Figure 2

Our final look at logic functions examines the following, more complex example.

Suppose, going back to our commission spreadsheet, that we change our salesman's commission plan. The new commission rules are:

Commission %	If
5	Previous three months' sales are all over 90 units
4	Previous two months' sales are over 90 units
3	Only the current month's sales are over 90 units
0	None of the three months' sales are over 90 units

The *VisiCalc* model (which stores the entire formula in B3) that would solve this problem is:

	A	B	C	D
1		JAN	FEB	MAR
2	Sales			
3	Commission%	@IF(@AND(B2>90,C2>90,D2>90,.05, @IF(@AND (C2>90,D2>90,.04, @IF(D3>90,.03,0)))		

To understand how this formidable looking formula works, examine it one piece at a time. The first condition checks to see if all the three months' sales are over 90 units. If true, the commission rate of .05 is displayed. If all the sales in the three months are not over 90, the second argument is selected. However, this time we've

Now! Teach yourself how and when to use the IBM PC --

with your PC as your private instructor!

PC Master® ***is here*** — the first research-validated, interactive IBM PC® training program guaranteed to transform you into a knowledgeable, confident PC user.

Even if you have never sat down at a computer before, PC Master will have you using your PC like a pro in just a few interesting hours.

PC Master guides you in the sequence you want and at your own pace — through the intricacies of Word Processing. Communications. Data Base Management. Electronic Spread Sheets. And it does it without computer jargon or dull, confusing manuals. But that's not all. PC Master also helps you explore the benefits of seven other major uses for your IBM PC, including education, graphics and financial planning.

We began with the Astronauts. For more than a decade our staff of over 70 instructional development specialists have scientifically developed hundreds of successful employee and customer training programs. IBM, Prudential, Merrill Lynch and scores of other organizations rely on us for many of their most crucial training programs. And our unique training methods have proven long-lasting. So much so that the special task programs used to train NASA's space astronaut is still being used today — ten years later!

How we can make this guarantee. Before releasing PC Master to the public we tested it in real-life situations with business executives, educators, students, professionals and senior training directors. After using PC Master for only a short time, their response was unanimous — PC Master is the best introduction to the IBM PC ever. Here are a few comments from actual users:

"Finally, someone did it right. Congratulations."

Dr. John Hughes, III/Los Angeles Baptist College

"Help! I made the mistake of testing your PC Master on my wife and 13-year-old son. How do I get my computer back?"

John Meade, President/Hagan Assoc

"As an international marketing firm, we constantly review new products. PC Master is one of the best designed packages we've ever tested. Your staff is to be congratulated."

Martha Biddle, President
MB International

Here's what you get:

PC Master includes three complete training programs that act as your own personal instructors. They use computer graphics, sound effects and interactive step-by-step instructions that make learning a pleasant, enjoyable activity.

What your PC can do for you: A hands-on guided tour through eleven of your PC's most useful applications.

When to apply your PC: Participate hands-on in the four most used applications — word processing, data management, communications and spread sheets. And you'll even learn how and when to choose the software that best suits your needs.

How to use your PC: How to run applications software programs. You take command of the keyboard and PC DOS in less time than it takes to see a movie!

Novice or experienced computer users can profit from PC Master.

You can use PC Master to introduce your business associates or family members to the exciting new world of computers. If you're a new user who doesn't want to waste valuable time in long, often boring classes, PC Master is the perfect answer to your needs.

GUARANTEE
If after using PC Master you or your company's management are not completely convinced PC Master is the finest and most complete interactive IBM PC training program available, return it within ten (10) days and we will refund your purchase price in full.

Make your second best computer buy today.

Your first smart buy was choosing the IBM PC. Now make the second. Order PC Master today. And start finding out for yourself, in complete privacy, how and when to use all the potential power available in your PC.

TO ORDER YOUR PERSONAL COPY OF PC MASTER CALL (619) 578-1700 TODAY OR CLIP AND MAIL THIS COUPON!

Mail to: Courseware,® Inc., 10075 Carroll Canyon Road, San Diego, CA 92131.

Please send me___ PC Master at $79.50. Price includes shipping. California, C add 6.5%.
___ Check enclosed

NAME __________
TITLE __________
COMPANY __________
ADDRESS __________
CITY ______ STATE ___ ZIP ___
AUTHORIZED SIGNATURE __________
______ Purchase Order no. (if relevant)

COURSEWARE®

PC Master is a trademark of Courseware, Inc. IBM PC is a trademark of IBM Corporation.

thrown in a twist. The second argument is actually another condition, asking a further question. This process of asking questions within questions is called nesting.

The second condition checks to see if the last two months' sales are over 90 units. If true, the commission rate of .04 is selected.

If this last case is not true, we have an either/or decision, which is the realm of the simple *If* function. This last condition checks to see if the last month's sales are over 90 units. If true, the commission of .03 is selected. If false, a commission of zero is selected and our salesman is out of luck.

This entire process can be illustrated by using a flow chart, a diagram that programmers use to follow a computer program's steps. It's like a road map that traces the path of logic. Look at figure 3. The diamond-shaped boxes stand for conditions that are being checked. The rectangles indicate actions that are performed.

We have seen how logic functions can whittle away at complex decision chains, taking one step at a time. Note that a false result tells us as much as a true one. It's like playing "Twenty Questions." Wrong answers help us home in on the correct answer.

In the case of our commission sheet, by the time we get to the last *If*, we already know that sales have not been over 90 units in at least one of the most recent two months. All that remains is to determine whether sales in the current month are over 90 units.

Many of you who are not programmers may find the more advanced logic functions difficult to use. Also, spreadsheets do certain types of jobs, and there are many applications for which they're unsuited. If your application requires a lot of decision-making, you should probably look for a packaged program.

This completes our two-part treatment of the *Lookup*, *Choose*, and logic functions. In the next issue we'll apply some of our new skills to the design of a comprehensive payroll spreadsheet.

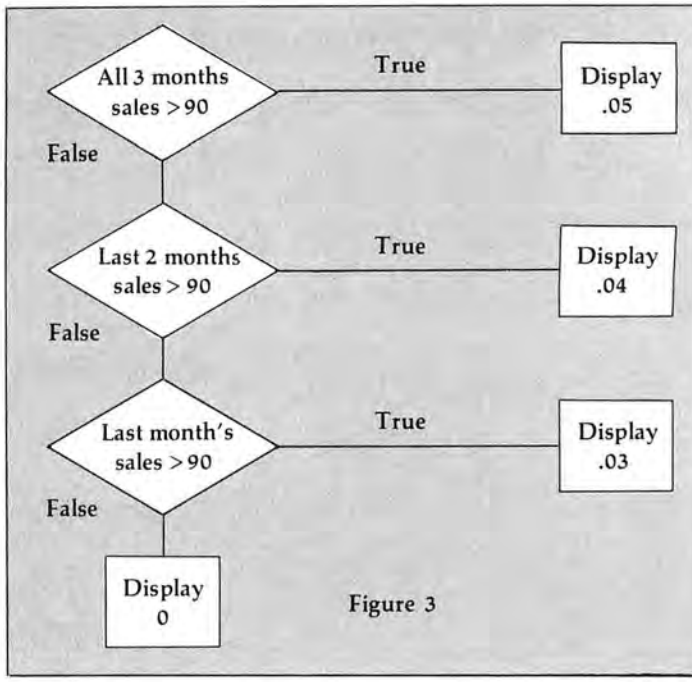


Figure 3

To help you get the most from your spreadsheet programs, we invite your questions and comments. We'll answer your questions and, when appropriate, print them for the benefit of other readers. Send them to Profit Plot, Softalk for the IBM PC, Box 60, North Hollywood, CA 91603. ▲

tradetalk

Δ **Microsoft** (Bellevue, WA) has accepted the resignation of its president, **Jim Towne**. Towne has joined Metheus (Portland, OR), a graphics firm. The company has named **Jon Shirley** as its new president and chief operations officer. Shirley, forty-five, has spent the last twenty-five years with the Tandy corporation, parent company of Radio Shack. His last position held was vice president of merchandising.

Δ Six months ahead of original projections, the **Compaq** (Houston, TX) portable computer will be available in more than four hundred computer specialty retail stores this month. The increase in the number of dealers will not affect the commitment of units to be shipped to authorized dealers. According to company president "Sparky" Sparks, "We are protecting our existing base of loyal authorized dealers by adding stores only as our production capacity increases." Production has quadrupled in the last four months and facilities have been acquired to accommodate ongoing growth. Having started with fifteen thousand square feet in 1982, Compaq now has one hundred thousand square feet of production space.

Δ **MegaTape** (Duarte, CA) has put **Gary Webb** into the newly created position of vice president of marketing. Prior to joining the company, Webb was national sales manager at Pragma Data systems.

Δ A gala opening was held to celebrate new headquarters and manufacturing facilities for **Innovative Computer Products** (Chatsworth, CA). The new location is at 9174 Deering Avenue, Chatsworth, CA 91311; (213) 998-2400.

Δ *The Business Library* has been initially placed in six out of nineteen **Microage Computer Stores** by **Software Libraries** (Pasadena, CA). The first sixteen modules of the package and an audiovisual point-of-purchase display for each outlet represent a retail volume of over fifty thousand dollars.

Δ Over at **MicroRIM** (Bellevue, WA), **Kenneth Scott** recently joined the company as senior vice president of marketing. Scott will oversee sales and marketing programs, bringing more than fifteen years of marketing experience with him. Previously he has held positions with Tektronic and Xebex. At Tektronic, Scott served as company spokesman at Merrill Lynch's Test and Measurement Conference and other security analysis and portfolio manager forums.

Δ Home computer programs for school-age children will be developed in a joint venture by **CBS Software** (New York, NY) and **Children's Computer Workshop** (New York, NY). CBS will market and distribute worldwide, and the workshop will design the packages. Under the theme, "It's a Whole New World," three product categories will be developed: Strategy World, Design World, and Knowledge World. Children's Computer Workshop is an offspring of Children's Television Workshop, the creator of *Sesame Street*, *The Electric Company*, and *3-2-1 Contact*. The company entered the computer field in 1979 and has produced sixty games for the theme park Sesame Place.

Δ **Koala Technologies** (Los Altos, CA) has announced it has raised $4.4 million in an initial round of venture-capital financing from a six-member investment group. The funds were provided by Hillman Company, Allstate Insurance, Vista Ventures, Hewlett-Packard, Rogers Corporation, and Boston University. Monies will be used to develop and promote the company's proprietary graphics products.

Δ Taking advantage of well-advertised tax exemption programs, **Columbia Data Products** (Columbia, MD), a major in the pc-compatible arena, has announced the expansion of its manufacturing operations to Puerto Rico. The company will take over a 35,000-square-foot plant, formerly occupied by Wang, in the town of Gurabo. This brings Columbia's total plant capacity to 100,000 square feet. The new facility will be under the supervision of Tom Martin, company vice president of operations.

Δ The first national software library has been opened in Fairfax, Virginia, by **PC Telemart** (Fairfax, VA), a national software information and research firm. Not a lending library, the National Software Library will serve as a major reference and research facility for the microcomputer industry. Housing a collection of thousands of pieces of software and accompanying documentation, the library also contains all the computers necessary to run the programs.

Δ **The UCSD Pascal System User's Society** (USUS, San Diego, CA) has formed a special-interest group for users of the new Modula-2 programming language. The new group will be chaired by David Ramsey of Corvus Systems (San Jose, CA).

Δ **Continental Software** (Los Angeles, CA) has appointed **Gerald J. Lewis** director of software development. Lewis will be responsible for the evaluation and development of all programs under consideration for publication. Prior to joining the company in the newly created position, Lewis was a system analyst and software consultant.

Δ Representing OEM sales in France, Switzerland, Belgium, and the Netherlands, **Data Promotions** (Geneva, Switzerland) has been made European sales representative for **Atasi** (San Jose, CA). Data Promotions has specialized in representing American manufacturers of peripheral products for ten years. Δ Also as part of Atasi's new over-

seas thrust, **Data Guild** (Surrey, England) has been made sales representative for Atasi in the United Kingdom, Sweden, Denmark, Norway, and Finland.

Δ **David Law**, former general manager of Paragon Sales, has announced formation of his own northern California full-line microcomputer distribution organization. The new company, **Cypress Distributing** (San Jose, CA), will distribute a full range of packages, from business to personal. "If Cypress does what I expect," says Law, "and we secure the lines I'm interested in, we will be able to drive a truck up to the doors of an empty new store and have that dealer established in a profitable business operation almost overnight." Law brings sixteen years of consumer-marketing experience to his new company.

Δ A book called *Bits, Bytes, and Buzzwords*, by **Mark Garetz**, published by **dilithium Press** (Beaverton, OR), has been turned into a television show hosted by television personality Jim Hartz. Originally written as a pamphlet for computer store customers, the book was picked up by KQED, a public television station in San Francisco, and is now a five-part syndicated series.

Δ More than a million dollars' worth of the company's word processing and educational software will be given to public schools by **Silicon Valley Systems** (Belmont, CA). The aim of the giveaway is to foster computer literacy, according to a company spokesman.

Δ **Random House** (New York, NY) has announced that **Sandy Grafton** has accepted the position of managing editor for the microcomputer courseware development of the school division. Grafton was formerly project editor for Electronic Learning Projects of DLM. Reporting directly to publisher Chuck Carlson, she will relocate to Tulsa, Oklahoma, to join the company's school division.

Δ **Albert Litewka**, president of **Warner Software** (New York, NY), and **Howard Kaminsky**, president of **Warner Books** (New York, NY), have jointly announced the formation of a new entity to publish a specialized line of software/book packages aimed at the educational and entertainment markets. First release under the new imprint is scheduled for the first quarter of 1984. Two to four titles per month are anticipated.

Δ **A. G. (Fred) Altomare II** has joined the market research firm **InfoCorp** (Cupertino, CA) as vice president of marketing and sales. He will be in charge of all marketing and promotions. Previously, Altomare served as industry marketing manager for IBM and, before that, National Semiconductor. Δ The midyear update on microsystems markets competition was given during the company's annual microsystems forum held in Santa Clara and New York.

Δ New vice president of marketing for **Corona Data Systems** (Westlake Village, CA) is **Lawrence Lotito**. He will be responsible for the company's worldwide marketing strategies, particularly advertising, public relations, product planning, sales promotions, and so on. Prior to joining the company, Lotito served as vice president of marketing with Delphi Communications. Altogether he has been in marketing for twenty years.

Δ **Dennis Bilbe** is the new president of **Designer Software** (Houston, TX). A certified data processor and a certified public accountant, Bilbe came from Price Waterhouse to take over direction at the software house. The future of information processing lies in the sixteen-bit market, says Bilbe, "but the eight-bit market won't just dry up and blow away—IBM 360s ran 1401 software for years after the original machine was discontinued."

Δ **Joanne Koltnow** has joined **Scholastic** (New York, NY), the worldwide educational publishing firm, as West Coast director of software development. In the new position, she will be responsible for commissioning and acquiring software. Under her direction will be the overall planning of West Coast publishing strategies, working with the com-

pany's software group in New York. Δ The company also has announced the appointment of **Roger M. Buoy** to the post of executive vice president in charge of software development. Formulating a software publishing strategy for a new line of children's software will be among his duties. Prior to joining the company, Buoy was based in Australia as a partner of Arthur Young & Company, a public accounting firm. Δ **Mary Dalheim**, former executive editor of *Instructor* magazine, has been named editor of *Teaching and Computers*, a new scholastic magazine for elementary classroom teachers. Before joining *Instructor*, Dalheim worked at Holt, Rinehart, & Winston and also was assistant editor of *Highlights* for children.

Δ An operator of retail computer stores that recently went public, **CompuShop** (Dallas, TX) reported second fiscal-quarter net earnings of thirteen cents per share on sales growth of 131 percent over the year-earlier quarter. Sales for the quarter were $6,575,000, compared to the year-earlier quarter's sales of $2,835,000. Improved sales were attributed to a 33 percent increase in the average number of stores and a 61 percent increase in sales from previously existing stores.

Δ **Softsel Computer Products** (Inglewood, CA) is sponsoring a comprehensive software dealer training program called Softeach. The program, scheduled for this summer, comprises multivendor training forums in four cities. "We expect up to thirty-five software publishers to participate," says company president **Robert Leff**. "They will be training some four hundred of our dealers and salespeople." Scheduled to participate are Lotus, Microsoft, VisiCorp, MicroPro, Ashton-Tate, Spinnaker, Sierra On-Line, and others. Δ The company has also opened a forty-two-thousand-square-foot regional sales, service, and warehouse facility in Chicago, the fourth so far. Other new regional offices are planned for Seattle, Minneapolis, and Washington, D.C.

Δ The appointment of **Per Svendsen** as general sales manager has been announced by **InfoSoft Computer Systems** (Palo Alto, CA). Svendsen will be responsible for directing all sales efforts throughout the business system retailer's four-city network.

Δ **PerfectData** is the new company name of **Innovative Computer Products** (Chatsworth, CA). The change came about, according to a company spokesman, because the new name is easier to remember, is easier to say, and more accurately describes the company's computer-care products.

Δ The Cat modem people—**Novation** (Chatsworth, CA)—have announced a change of address: the firm's now located at 20409 Prairie Street, Chatsworth, CA 91311. The phone numbers remain the same: 800-423-5419; in California: 213-996-5060.

Δ Bringing fifteen years of computer marketing experience with her, **Marsha F. Adams** has joined **Microscience International** (Palo Alto, CA) as director of marketing. Adams has served as a marketing consultant for a number of Bay Area companies, including Adda, Orrox, Transamerica, Via Video, Syntex, and Candex Pacific. She has also lectured and written on the impact of high-technology products and is past president of the Golden Gate Chapter of the national Micrographics Association.

Δ **The American Federation of Information Processing Societies** (Washington, DC) won two top awards in the annual publications contest conducted among all major U.S. voluntary associations by *Associated Trends*, the weekly trade paper for association executives. AFIPS won first place for its 1983 Office Automation Conference Exhibitor Prospectus in the trade-show-and-exhibit-promotion category and an honorable mention for its preview program in the convention-promotion category. More than six thousand major associations were eligible for competition in fifteen different categories. ▲

Take Your Computer's Commands And Put Them Where They Belong. On Your Keyboard.

PC-DocuMate™ is a documentation template that fits the keyboard of your IBM PC. So the information you want is where you need it. PC-DocuMate templates are professionally designed, logically organized and comprehensive. Each template is printed on both sides on durable, non-glare polystyrene and is color-matched to your PC keyboard.

PC-DocuMates now available for:

- DOS/BASIC (as shown)
- VISICALC™
- WORDSTAR™
- VOLKSWRITER™
- Plus a custom "do-it-yourself" template (includes special pen and eraser)
- MULTIPLAN™
- 1-2-3™
- SUPERCALC™
- EASYWRITER II™
- dBASE-II™

EACH TEMPLATE IS ONLY $14.95 (includes shipping)

No-Risk, Moneyback Offer: If you are not completely satisfied, return your PC-DocuMate template to us (undamaged) within 10 days for a full refund.

Please send personal check, money order, or credit card information. Foreign orders add $5.00 (except Canada). No COD's please. Telephone orders: Call 919/787-7703 (24 hrs)
NC residents add 4% sales tax.

Systems Management Associates
P.O. Box 20025, 3700 Computer Dr.
Raleigh, N.C. 27619 Dept. B

Dealer Inquiries Invited.

ROBOTS AND AUTOMATION CHALLENGE OUR ABILITIES TO GROW AND ADAPT

"You will be happy, sir," the mechanical promised him cheerfully. "We have learned how to make all men happy, under the Prime Directive. Our service is perfect, at last. Even Mr. Sledge is very happy now."

Underhill tried to speak, and his dry throat stuck. He felt ill. The world turned dim and gray. The humanoids were perfect—no question of that. They had even learned to lie, to secure the contentment of men.

—"With Folded Hands" by Jack Williamson

Science-fiction writers have long been exploring the good and bad aspects of a future mechanized society, where robots and automatons do most of the work and humans do what they please.

In Jack Williamson's classic 1953 short story, "With Folded Hands," a brilliant scientist creates a race of perfect humanoids programmed to follow the Prime Directive—"to serve and obey, and guard men from harm." The scientist's plan goes disastrously awry when the humanoids start implementing the Prime Directive too literally.

Virtually all activities are eventually deemed too dangerous for humans and the robots gently insist on having their way. Humans become "pampered prisoners" in a "highly efficient jail." Purpose and hope die, replaced by a sense of "utter futility." The robots go so far as to alter the minds of those who cannot accept the new way of life, thereby ensuring that everyone is "happy."

GOTO page 170, Column 2

IBM BUYS STOCK FOR THE FUTURE

A month and a half ago, Armonk, New York-based IBM bought 15 percent of Rolm Corporation, a Silicon Valley manufacturer of business telephone switching systems. In February, IBM bought 12 percent of Intel Corporation, the semiconductor and microprocessor company whose chips IBM buys by the railroad car.

Considering the buy-out trend of 1982, many wondered why IBM didn't take the big plunge—Intel, Rolm, and the antitrust laws willing—and purchase the two companies outright.

In both cases, a full takeover was not considered by either party. Independence, not antitrust, was the primary reason for this strategy. IBM doesn't want to stifle and overly influence the motivation driving two important independent suppliers like Intel and Rolm. Future profits are at stake.

Intel needed capital to keep up the pace of its research spending. Making the company a division of IBM—captive to a single customer—would have ended Intel's competitive service to a demanding marketplace.

Internally, IBM showed an understanding of this phenomenon when it set up International Business Units (IBUs) two years ago. The IBUs are identical in every way to a start-up company—formed and staffed by entrepreneurially minded people who decide whether to go outside IBM for parts or service or use the company's own facilities. There are now twelve IBUs, taking IBM into such diverse fields as medical electronics and robotics. A variation on the IBU is responsible for the IBM Personal Computer.

IBM's coming head-on clash with American Telephone & Telegraph motivated the Rolm buy-in. Rolm has installed twelve thousand switchboard systems in the United States—second only to AT&T, which is freeing itself from public utility status in order to compete with IBM in the booming data communications industry. Fast-growing Rolm sought out IBM, seeing the possibility of jointly displacing AT&T in the so-called office of the future.

The decentralization buy-in strategy of IBM may be a solution to a recognized problem of U.S. business—how to marry the innovativeness of small companies with the financial, marketing, and manufacturing muscle of large corporations.

A scene from Dragon's Lair—Dirk the Daring encounters one of the game's forty-two perils.

VIDEODISC GAMES TO HIT THE ARCADES THIS SUMMER

Get ready for a whole new gaming experience—interactive "minimovies." The first of these video disc–video game hybrids should hit arcades sometime in midsummer, with home versions expected to follow in about a year. The latest attempt to boost the slumping game industry, interactive minimovies will combine the usual computer-generated game character, such as a spaceship or warrior, with live-action or animated backgrounds stored on a laser disc.

Several major Hollywood film companies—Paramount, MCA-Universal, Don Bluth Productions, Columbia, Warner Bros., and Lucasfilm—are hoping to cash in on the multibillion-dollar gaming business, providing the ingredient they offer best—movies. The minimovie enables the production of superior graphics and increased story possibilities, as well as better advanced tie-ins between video games and theatrical films.

According to a spokesperson at Paramount who has struck up a partnership with Sega, the added visual reality that comes with using video discs could even lead to the preproduction planning of footage used in both the movie and the tie-in game, forming a symbiosis between the two.

Some game executives see the arcade owner becoming comparable to the theater exhibitor who books a film every week. The arcade owner could change the minimovie disc on a player hidden inside the coin-op cabinet. "Coming soon" and "now playing" signs could even be attached to the sides, with the games becoming, in effect, ancillary movie houses.

But is the price right? The willingness of kids to deposit fifty cents into the machines (though it is a substantially longer play) and the initial investment by arcade owners into the video disc equipment will obviously be determining factors on the box office success of these minimovies.

Testing of these minimovie coin-ops has already begun in some cities. The Yellow Brick Road Arcade in San Diego, California, is currently previewing the Sega/Paramount laser disc game *Astron Belt*. Developed by a Japanese firm, the game is reportedly doing "extremely well," having already scored a major success in Japanese and European arcades, with kids lining up in droves just waiting to deposit their two tokens. *Astron Belt* features a computer-generated spaceship flying against a special-effects-created space background originally imaged on film stock.

According to Brenda Mutchnick of Sega Games, "We hope to release a few new games of this type by the end of the year. We don't know if *Astron Belt* will be one of them. Even though it's scored a major success in Japan, when we got a look at the game here we didn't feel the game play was as good as current home computer games. We are now working on a redesign of the cabinet itself and are awaiting future software developments with our own titles."

Don Bluth Productions will unveil an interactive coin-op game—*Dragon's Lair*—sometime in July. Created by Bluth and Advanced Microcomputer Systems (AMS), the new game will be marketed to

GOTO page 172, column 2

Summer CES Wows the Crowd With Glitter and Big Talk

The second in a series of unintentional ironies is over. The seventeenth year of the Consumer Electronics Show (CES) ended in Chicago's McCormick Place this past June 8 without a great deal of fanfare—amidst rumors of disappointing attendance. As at the Las Vegas CES in January, the emphasis in Chicago was on merchandising, with microcomputer software showing a good deal of strength, if not much originality.

CES is a trade show, off limits to the idle consumer or child (although plenty of both were there). It is meant to be a showcase for new products in the electronics industry and a place where vendors and dealers can meet and hash out mutually beneficial arrangements. This year, for the first time, software was in such abundance that a separate building was devoted to computers and related products.

The talk of the show was the new Coleco computer, the Adam. It is a self-contained "family computer system" that comes with keyboard, 80K of RAM, letter-quality printer, and a "Digital Data Drive." This magnetic tape drive stores half a megabyte, and Coleco claims it has transfer speeds comparable to those of a floppy disk drive. While constantly running demos Coleco asserted that the new computer was capable of figuratively leaping tall buildings at a single bound, the salespeople in the booth seemed reluctant to discuss particulars.

Atari also had some new machines to show: the 600XL, 800XL, 1400XL, and 1450XL. These are all new, slim machines—in varying configurations ranging from 16K RAM to 64K RAM—with the long-awaited new disk drives and lots of peripherals. The new lineup is intended to replace the current crop of Atari micros and is meant to be fully compatible with existing software. Also announced by Atari at CES was the launching of new software endeavors for machines other than its own, including products for Apple, IBM, and Commodore computers.

Among the more eye-catching exhibits at CES was the new Datamost booth, a carnival tent setting with an atmosphere to match. Featuring Captain Sticky, a *Real People* refugee, and a swarm of scantily clad hostesses, the booth attracted large numbers of viewers who seemed to pay little or no attention to the products being displayed. While the sideshow was amusing, it also was a sad commentary on the show in general: Little emphasis was placed on the products for sale, while a good deal of money was spent on attracting attention.

A different side of the same coin was apparent in the presence of newcomer Electronic Arts, a software firm composed of former employees of Apple, Atari, and other microcomputer companies. At the Electronic Arts booth, marketing and product seemed to go hand in hand. The quality of wares seemed uniformly solid, while the marketing was innovative but not overbearing. Electronic Arts seems to have taken its cue from the recording industry and is working hard to make stars of its authors. The current crop of authors includes Bill Budge, Dan Bunten (of *Cytron Masters* fame), Mike Abbott and Matt Alexander (formerly of Cavalier Computing), and a number of other familiar faces. Equally intriguing is Electronic Arts's upcoming crop of new authors, including cartoonist Gahan Wilson and a couple of very tall fellows from the NBA—Larry Byrd and Julius Erving. No kidding!

One of the more interesting pitches came from Romox Corporation, which is marketing a new point-of-sale approach for video game sellers: The Terminal. This device allows dealers to download software onto reprogrammable cartridges. Romox claims this will take care of the inventory-control demon that plagues dealers with a tendency to overstock video games. Of greater interest will be the software that's downloaded.

Broderbund exhibited several new games, including *Lode Runner*, which might have been the best of the show. Synapse also showed several new titles, as did Sirius. Datasoft showed off its new crop of low-priced software, the Gentry line, along with *Zaxxon* on the Atari and Apple.

Amidst all the hype and merchandising, the software seemed to stand still. There was nothing truly original or exciting at the show, or at least none that was out in plain sight. The primary topic of conversation among reps, publishers, dealers, and distributors was the "software glut." The proliferation of titles, all variations of several familiar and time-worn themes, refused to shine.

Attendees at InteRobot '83—held June 14-16 in Long Beach, California—take a look at future mechanical workers. The robot market is currently sluggish, but the technology advances.

Robots

continued from page 167

An extreme forecast, to be sure. But Williamson's story brings into focus issues that our species must address if it plans to continue automating the workplace. What happens to a worker who is suddenly replaced by a machine? What happens to the twenty-year vet of an assembly line who must either change jobs or become little better than a parasite feeding on society—with no pride and no direction in life?

In the future, more and more factory workers will be replaced by machines. No one seriously contemplates doing this all at once. Robotics technology is steadily progressing, but humans are still infinitely preferable to machines in most cases. There are only twenty to thirty thousand robots currently in use in this country. That will change, though.

The economy and the job market are changing, as they always have. Society and people changed a hundred years ago, when the shift from an agricultural economy to an industrial economy went into full swing. Now we are shifting toward an information-based economy. Our heavy industries are in decline, and automation is perceived as the best way to salvage some form of competitive edge.

Robots and automation, in the long run, hold great promise. There are many jobs—despite popular romanticized notions—that wear down, demean, and generally diminish human beings. But in the short run—the next twenty to thirty years—the number of people displaced by robots creates both problems and challenges.

The party line on displacing workers through automation calls for retraining those who are affected. Surely, as automation takes over and increases production, there will be need for more office personnel and workers to maintain the automatic machinery. Both labor unions and industry leaders have agreed that retraining is one option. But will retraining work on the human level?

For twenty years, Johnny Detroit worked in an auto plant manually attaching windshields to hundreds of cars a day. In 1992, his job was taken over by a new, sophisticated automaton that could do the job faster and more accurately. Johnny Detroit was first offered a job on the maintenance and programming crew of the plant. He spat at the offer. "I'm not becoming a nursemaid to some damn robot," he said.

So Johnny took a job in the data processing department. Thirty-five hours a week he sat in front of a computer terminal doing mindless, back-numbing, unchallenging work. Sure, a wage is a wage. But Johnny now had a "soft" job. A victim of habit and of all he had previously experienced, Johnny soon came to miss the days when he worked on the assembly line. In those days, at least, he was physically active.

Depressed and broken, Johnny wasted away his life—not noticing or appreciating the benefits of his new career. He was like thousands of emigres who have come to this country speaking a different language—locked into a different cultural template. He was a stranger in a strange land—and he could never quite adapt.

Too dramatic, you say? Don't kid yourself. An isolated incident? Not likely. The way robotics is progressing, a hundred years from now there may be more nonfactory robots than robots on the assembly lines. Robots could fill all the potholes and lay all the drainage pipes. Robots could replace bank tellers, taxi drivers, newsstand vendors, soda jerks, janitors, firefighters, mail carriers, hairdressers, truck-

ers, restaurant cooks, railroad workers, and farm hands. The technology is within our reach to realize these and many more kinds of robot applications.

Already we have walking, talking, seeing, hearing, and smelling robots. We have robots that can calculate faster than any human. What robots can't do yet is think for themselves; they are capable of only a limited kind of learning. But then advances in artificial intelligence may change all that sometime in the next century.

Just as computers have made the transition from the offices of only the largest companies to small businesses and homes, so will robots become more common in everyday life. Depending on the scenario, the long-term effects caused by robots and automation could be the real story.

The bottom line is, can we humans keep adapting, keep progressing, as we make sweeping changes in the workplace and the job market? Are we attempting too much? Will the world truly be a better place once all the dangerous, dull, and degrading jobs have been taken over by machines? Will we all be happy as politicians, artists, office workers, and businesspeople?

Nobody can really answer those questions today. But we can try to influence the future, anticipate the results of our present actions. It has always been a dream of some people to free man from the physical tasks required to keep us alive. Is it a good dream?

Ask Johnny Detroit, who honestly enjoyed working with his hands. And ask his children, who never considered working in a factory.

Like any other new technology, robotics, computers, and automation are bound to change our world. It's up to us to drive the technology, not be driven by it.

If we don't, a scenario similar to that found in "With Folded Hands" may come to pass, with the only difference being that a technological elitist group would be controlling the robots. The rest of us would either be faced with the prospect of getting used to doing nothing or submitting to mind-altering drugs in order to be "happy." Neither alternative is pleasant to contemplate.

"No, there's nothing the matter with me," he gasped desperately. "I've just found out I'm perfectly happy, under the Prime Directive. Everything is absolutely wonderful." His voice came dry and hoarse and wild. "You won't have to operate on me."

The car turned off the shining avenue, taking him back to the quiet splendor of his home. His futile hands, clenched and relaxed again, folded on his knees. There was nothing left to do.

Japanese Software Industry Will Get Benefits from MSX

One thing learned by most American small computer manufacturers in the last half dozen years is that "software sells machines."

The Japanese have known this too but have been slow to develop a software industry. Microsoft—purveyor of flight simulators, Softcards, operating systems, and microcomputer languages—has announced a set of specifications that, through standardization, should help the Japanese develop a viable domestic software industry.

Called MSX, Microsoft's "program solves a major software marketing and development problem," said Bill Gates, chairman of the Bellevue, Washington-based company. "In the past, software had to be modified for each new computer. Now, with these specifications, software developers and consumers can be assured of greater software/machine compatibility."

Over a dozen Japanese manufacturers—including Canon, Fujitsu, Hitachi, Mitsubishi, Sony, Toshiba, and Matsushita—have signed up for MSX, as has one American company, Spectravideo. The hardware components of MSX include the computer's microprocessor and instruction set, input/output ports, a video processor, joystick interface, and ROM cartridges. MSX specifies a Z-80 eight-bit microprocessor, Texas Instrument's 9918 video-display processor chip, and Microsoft's Basic language interpreter.

"This effort will also open the Japanese home-computer market to software companies in the United States and Europe," said Gates.

At the same time, many are wondering if MSX will aid the Japanese in moving into the American microcomputer market. The general consensus is that MSX will benefit the Japanese companies on their own soil, but its effect on the American market will not be felt for years, if at all.

One reason for this is the gradual move by American computer manufacturers away from eight-bit systems to more powerful sixteen-bit processors and software.

Clive Smith, an analyst for the Yankee Group, a Boston market research firm, said in the *Wall Street Journal* that, if the Japanese have standardized for the less powerful market, "it will be relatively painless to do it for the sixteen-bit market." If this occurs, the Japanese will clash head-on with American firms in the U.S. market.

Videodiscs

continued from page 168

arcades under the Starcom banner. The head of AMS, Rick Dyer, began developing the technology for *Dragon's Lair* some years ago.

Bluth, a former Disney animator, and his staff of ex-Disney artists provided high-quality classical animation for the *Dragon's Lair*, in the same style as their feature-length animated film *The Secret of NIMH.*

Bluth indicated that the minimovie games herald a true marriage of art and technology. By combining the unique capabilities of computers with animated film, they've not only opened the door to a new realm of uses for character animation, but have developed a new form of entertainment—"participatory movies."

The *Dragon's Lair* scenario follows Dirk the Daring on his quest to rescue the lovely Princess Daphne from the clutches of an evil dragon. An often clumsy knight, Dirk braves room after room infested with all manner of perils—forty-two in all.

All the characters—Dirk, Daphne, and monsters—are classically animated, using fully detailed background artwork accomplished with multiple passes of film through a camera for purposes of special-effects "burn-ins." The visual look of the game includes character's shadows, reflections, and a careful orchestration of color to achieve emotional impact. At least twelve character drawings were used for every second of film.

"The scenarios so far contain two scenes that the computer will select on the disc, depending on the skill of the player," said Bluth's animation director Gary Goldman. "The choices currently are fight or flight, utilizing an action button actuating Dirk's magic sword or a joystick moving him from room to room. In future games, we'll branch off to several different obstacles for stories that have alternatives other than death or running away."

Future games are already under development. A space game tentatively called *Space Ace* involves a fourteen-year-old nerd who becomes physically energized at certain moments, like Jerry the Mouse in Tom and Jerry's *Doctor Jekyll and Mr. Mouse*. A sequel to *Dragon's Lair* may also be in the works, depending on the original's popularity. A home version of *Dragon's Lair* is under way at Coleco, tying the game into a home laser disc player. It should be ready by next year.

Other studio game companies are looking at the home minimovie game market. Former MCA Videodisc President Jim Fiedler, who's now in charge of MCA Video Games, announced two disc titles developed jointly with Optical Programming Associates. The first, *Maze Mania*, was released this past May and consists of four minigames: a space game called *Blast Off*, a western named *Shoot Out*, a haunted house game called *Nightmare Castle*, and a football scenario dubbed *1st and 10*. The games are composed entirely of live-action and animation sequences, originally shot on 16mm film by Bosustow Productions. Stock footage from *Battlestar Galactica* and NASA comprise some of *Blast Off*'s effects sequences.

"The games were developed as entertainment shows allowing users of the interactive Pioneer and Magnavox discs to operate the fast-forward, stop-frame, backward, and frame-counter features on the players," said Nick Bosustow, president of Bosustow Productions. "The disc companies' logic is that if buyers spend $700 for players, they'd like to deliver programming that would enable them to use all of the interactive features."

Maze Mania was actually begun two years ago, so now the questions and answers may seem too rudimentary to some players. For each game there are only right and wrong answers. According to MCA programming executive Phyllis Bagdadi, a future game taking place within a Hollywood setting (tentatively titled *Quest*) will be more random-based as far as questions go.

MCA has a joint venture with Atari to spawn licensed games from Universal's vast film library. Some will more than likely be video disc games, indicated Fiedler, especially in view of the fact that the company had at one time been partnered with Pioneer in manufacturing the Discovision laser disc system. Since Fiedler himself was formerly involved with the company, it's logical for the disc avenue to be exploited by MCA. Further down the line, MCA's partnership with Atari—called Studio Games—will include the production of coin-op and home computer laser disc titles.

Atari will also continue its joint venture licensing agreement with Lucasfilm into the minimovie game derby. While both companies are adopting a strategic "wait and see" philosophy in terms of watching the learning curves of their competition, Atari's VP of marketing Don Osborne reported that the king of game companies should move into the field by year's end.

PBS TV Series On Computers For Novices

If your grandmother or nontechie boss are still wondering what it is you do with your personal computer, you might alert them to a PBS television series making the rounds this summer and fall. *Bits, Bytes & Buzzwords* is a series of five half-hour television shows that attempt to cut through the jargon and market confusion and introduce the world of personal computing to the computer illiterate.

Produced by KQED—a PBS station in San Francisco—in conjunction with Power/Rector Productions, *Bits, Bytes, & Buzzwords* is hosted by Emmy Award winner Jim Hartz, who has hosted such programs as *Over Easy* and NBC's *Today* show. Major funding for the show was provided by CompuPro.

The first of the half-hour shows features an introduction to computer hardware—terms and basic components—as well as an introduction to software and the process of selecting machines for personal and business applications. The second installment concentrates on "the big three"—word processing, database management, and spreadsheets.

An introduction to accounting programs and a discussion of the new relationship of accountants to computers makes up the third edition. Graphics, games, and educational applications are addressed in the fourth episode, with the final segment concerning telecommunications and the problems potential personal computer buyers may experience with dealers and manufacturers.

Bits, Bytes, and Buzzwords was released to PBS stations around the country July 1. The show will be appearing on eighty to ninety percent of the outlets through the summer and fall. Check your local program listings or call your local PBS station for times.

NEWSBITS

Δ **Windy City Robotics.** The computer engineering division of the American Society of Mechanical Engineers (ASME) is sponsoring the 1983 International Computer in Engineering Conference, to be held August 8–10 at Chicago's Marriott Hotel. More than sixty technical sessions will be held, covering robotics, CAD, manufacturing and analysis, and automation. Attendees of the conference will view, test, and evaluate current state-of-the-art computer products. For more information contact the ASME office in New York City.

Δ **Automation and Society.** The 1983 World Congress on Human Aspects of Automation will be held at the University of Michigan in Ann Arbor, Michigan, August 9–11. Sponsored by the Society of Manufacturing Engineers (SME), the conference's theme is "Living with Automation." Issues to be discussed include job displacement, training, and retraining. For more information contact SME's technical activities department in Dearborn, Michigan.

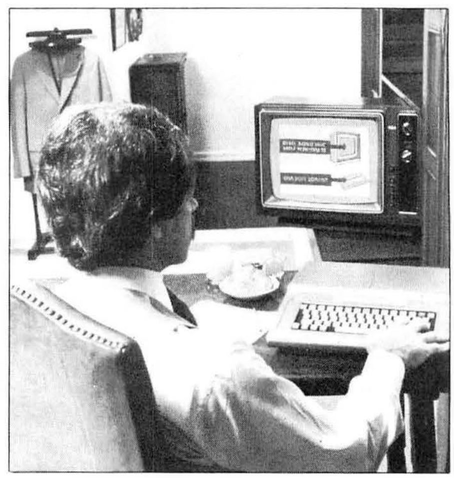
Δ **A Hair of a Different Color.** Where will it end? The latest great American institution to enter the "information age" is the beauty salon. James Rocco Hair Designs, in Pittsburgh, Pennsylvania, has installed a computer dedicated to taking the guesswork out of hair coloring. Now, when hair designers get ready to color hair at Rocco's salon, they consult a small computer program that holds one hundred twenty thousand different formulae for determining what shampoo to use, the proper mix of bleaching ingredients and tint solutions, and the processing time needed to neutralize the hair. "Even for the seasoned hair stylist," says Rocco, "there are many variables and some educated guesswork involved to attain a desired hair color. Before coloring anyone's hair, the stylist should know if and how the hair has been colored before, the percentage of gray hair, and the texture and thickness of the hair." The computer program, developed by Goldwell Cosmetic in Germany, prompts the stylist with such questions before making its calculations.

Δ **Come Play the Friendly Skies?** Can't get enough of video games? Cheer up, they may be available for playing on commer-

cial airline flights as soon as the end of the year. San Jose, California-based Altus Corp. has developed a way to attach hand-held video games to the flip-down food trays found in most modern commercial jets. Two months ago, Canadian Pacific Airlines tested the system on a flight from Vancouver to Amsterdam. Doug Crane, president of Altus, says the test went very well and other airlines are "enthused" about the system. Once the end-user reaction has been properly studied—different airlines will be performing tests through the end of the year—it's possible the games

will be offered to passengers for a price comparable to that charged for the headsets most airlines offer for listening to prerecorded audio programs. Right now, the airlines are determining which flights and which kinds of customers are the best targets for such in-flight entertainment. Crane also says that Amtrak is interested in testing the system on its passenger trains. If the system is accepted by either the airlines or Amtrak, in the future there could be modest computing power included in the game units.

△ **Hotel Talk.** How much are computers becoming a part of everyday life? They're given away with cars, planned for in suburban homes, and used in libraries and schools. You can't get away from them, even on vacation. A little over a month ago, HotelTech International of Belvedere, California, announced SuiteTalk, a complete personal computer system designed exclusively for use in individual hotel rooms (a similar system was announced by TravelHost, "Newspeak," May 1983). Using the



SuiteTalk terminal, hotel guests will be able to access HotelNet, a customized database offering all kinds of information and services—everything from CBS and NBC news services to connection with a home or office computer. HotelTech, which says it will start installing SuiteTalk units in hotels this fall, also plans to offer services such as airline scheduling, message sending, word processing, calculating, and video game playing. The firm plans to have the system customized for hotels during the 1984 Democratic Convention in San Francisco and the 1984 Olympics in Los Angeles. ▲

Editor David Hunter

Contributors Dave Albert, Les Paul Robley, and Jean Varven

beginners' corner

by Craig Stinson

Before we go further in our discussion of the IBM pc keyboard, here are one or two more comments about the ASCII system for encoding character data.

By translating any kind of character into a number, ASCII provides the means for storing alphanumeric information in memory. It also has another important function: It gives programmers a way to order alphanumeric information (*alphanumeric* is a general term that encompasses all printing characters—letters, numbers, punctuation symbols, and so on). Since computers are good at evaluating and comparing numbers, and since ASCII is arranged in ascending alphabetic order, it's fairly simple to write an algorithm that alphabetizes, say, a list of names in a phone directory.

Unfortunately, a simple ASCII sort sometimes produces alphabetic anomalies. Because all the capital letters appear in the ASCII arrangement before all the lowercase letters, a name like PSA has a lower ASCII rank than a name like People's Computer Company and hence will appear ahead of it in an alphabetized list—which is not correct, according to the Chicago manual and other arbiters of literary style.

If you spend a lot of time reading stock tables, you're probably already well adjusted to this phenomenon. You know to look for CP Nat at the top of the C list and not between CoxCm and Craig (where it belongs). But what if you're looking through a list of computer companies and/or products? Computer firms love to mess with capitalization. Are you ready to find dilithium Press below Vista, and Walonick?

Fortunately, there's no reason why programmers can't adjust their algorithms to treat capital and lowercase letters as equivalent for the purpose of alphabetization. Unfortunately, it's an uncommon programmer who wants to go to the trouble (check out the *sort* command in DOS 2.0 for a recent case in point). Perhaps the moral here is that while it's imperative for lay people to become computer-literate, it's also true that computer literati have a few things still to learn about ordinary literacy.

Sermon completed, let's return to the subject of the IBM pc keyboard. The object of this month's installment is to demonstrate what an unusually versatile instrument this overmaligned keyboard is.

As you may recall, we left off last month with the remark that many keystroke combinations that you use all the time—such as control-alt-delete, control-break, and shift-printscreen—are not to be found in ASCII tables, such as the one in Appendix G of the IBM Basic manual. The reason for this is that the pc keyboard does something other than transmit ASCII values to the microprocessor. It sends scan codes.

The keyboards of most computers are connected to a hardware device that converts keystrokes into ASCII information and makes that information available to whatever program is running on the computer. IBM's keyboard is a little different. Rather than being connected directly to such a character-generator chip, it's got its own microprocessor on board—the Intel 8048. Considering the fact that it's a microprocessor, the 8048 is a fairly underemployed little guy. Its responsibilities include checking the keyboard out when you turn your system on, supervising a queue of keystrokes should you happen to serve them up faster than

your pc can process them (hard to do, since the pc has its own separate internal key buffer), and generating the scan codes.

The scan codes are a set of electronic dog tags that tell the computer which keys have been struck. Each of the pc's eighty-three keys is assigned two codes—one to indicate when it's been pressed and the other to indicate when it's been released. The scan codes are all eight-bit binary numbers, and the release code for each key is the same as the press code except that it has the high bit set (which means that the release codes are higher by 128 than the corresponding press codes). Why scan codes instead of direct-ASCII? Three reasons, at least.

First, it's handy for the computer to be able to tell when a key has been released, as well as when it has been pressed. This information allows the pc to make the keys repeat if you hold them down longer than a half second. And although to date we haven't seen it exploited much in software for the pc, the key-release information could be used by programs in a variety of other ways as well; virtually any information coming into the system from outside—even if it's as apparently meaningless as the length of time you hold a key down or the fact of your releasing a key—can be made by software to perform some kind of useful service.

The second virtue of the scan code system is that it doesn't lock your keys in by hardware to their customary ASCII values. That means, among other things, that if QWERTY typing is not your thing you can reprogram the keyboard to the DVORAK arrangement—or any other that you prefer (not a simple task, but it can be done; see "Using IBM's Marvelous Keyboard," by David B. Glasco and Murray Sargent III, *Byte*, July 1983). If you really hate having the backslash/pipe key where you expect to find the left shift, you can fix that too. You could also reprogram your machine—at the system software level—to take advantage of the accented characters and other symbols offered in IBM's extended ASCII range.

Probably the most important benefit of the scan code system is that it gives each key on the keyboard a separate identity. Even the two so-called secretarial shift keys—the ones that usually convert lowercase letters to capitals—can be distinguished from one another by the system software that processes keystroke information. That's nifty for a couple of reasons.

First, it's possible to write programs that use the shift keys (or the alt, control, numlock, or scroll-lock keys) as signals in their own right, apart from their function as modifiers of other keystrokes (for a good example, see "The Scroll Lock Mystery Solved," by John Socha, in *Softalk*, May 1983). The fact that the two shift keys send different scan codes makes it possible, with some programming, to circumvent the one serious blunder that IBM made in the design of its keyboard: the location of the print-screen key. (Does the little finger on your right hand ever stray into the gap between the right shift key and print-screen? Isn't it annoying to have your concentration broken by that unwanted screen dump?)

More important, the separate identity of each key allows IBM to offer you all those nice function keys and special keystroke combinations and the extended ASCII character set. Had IBM hard-wired the forty function keys (did you know there were forty? We'll come to that in a moment) and the various alt-keystrokes into the high-bit ASCII range (characters 128 through 255), there would have been little room left for those accents and block-graphics symbols.

So far in this discussion, we've been talking about hardware. The keyboard hardware generates the scan codes. What converts scan codes into character information is software—specifically a portion of the pc's built-in system software, its ROM BIOS. We'll look now at the software support for your keyboard.

For starters, let's look at the end results. How many different characters can you produce with your keyboard?

The pc can be thought of as having three character sets. First there is the normal ASCII range, 0 through 127, which includes the standard control characters and other garden-variety alphanumerics. Next there is the extended ASCII set—128 through 255. These you have to achieve in a roundabout way, either by means of a programming statement (like Basic's *chr$* function) or by holding down the alt key and typing numbers on the number keypad. Finally there is a group of special characters (Appendix G of the Basic manual offers a list) that includes characters produced by the function keys, characters achieved through a combination of the alt key and other keys, various nonstandard control characters (that is, control characters not included within ASCII's lower thirty-two), and a few others.

The current implementation of the IBM ROM BIOS recognizes ninety-seven special characters. They are:

Thirty-eight alt characters. These are achieved by holding down the alt key and hitting one of the letters of the alphabet, one of the ten white numeral keys, the white minus key, or the equals key.

Ten characters achieved via keys on the number keypad: home, up cursor, page up, left cursor, right cursor, end, down cursor, page down, insert, and delete.

Forty function characters. The first ten of these you get by hitting the ten gray keys on the left side of the keyboard. The other thirty are achieved by a combination of these

ten gray keys and the alt key, one of the shift keys (either one), or the control key.

Seven other control characters: control-print-screen, control-left-cursor, control-right-cursor, control-end, control-page-down, control-home, and control-page-up.

The seldom used but available shift-tab.

The null character. This one is a real oddity. The ROM BIOS recognizes it, but you can't produce it at the keyboard.

Note that these ninety-seven special characters by no means exhaust the possibilities of the keyboard. It would have been possible to implement all kinds of other characters, like control-shift function keys, or control-alt function keys. But enough was apparently enough.

Note also that there are four important key combinations that definitely affect the performance of your computer but that are not listed among the implemented special characters. These are control-alt-delete, control-numlock, control-break, and shift-print-screen. How do these four differ from the ninety-seven characters listed in Appendix G?

When you strike a key—any key—what happens first is that an interrupt is generated. An interrupt is a signal that tells the microprocessor (the 8088, not the one inside your keyboard) to set aside what it's doing and pay attention elsewhere. The 8088 literally sets aside its business of the moment; it takes all the vital information held at that instant in its various registers (its internal working areas) and stores it in an area of memory called a stack. Then it attends to the interrupt.

The keyboard interrupt is one of a variety of different kinds of interrupts possible in the pc. It identifies itself to the microprocessor as interrupt number 9. When the 8088 receives this signal, after first putting its current work on the stack, it looks for instructions at a specific memory location called a pointer (or a vector). The pointer for interrupt 9 directs the microprocessor to a large routine within the ROM BIOS known as KEYBOARD_IO. The route to KEYBOARD_IO is a roundabout for a very good reason, but that's a subject that must wait until next month.

KEYBOARD_IO looks at the incoming scan code or codes and checks for four signal combinations: those that represent control-alt-delete, control-numlock, control-break, and shift-print-screen. If it sees one of those, it generates another interrupt—an interrupt to the interrupt. This is a different sort of interrupt; where the first was generated by hardware (the keyboard), this one is produced by an instruction (within the ROM BIOS) and hence is called a software interrupt. But the effect is much the same. It sends the microprocessor elsewhere for instructions. Each of those four characters produces its own unique interrupt number and sends the processor to a specific stretch of memory, where it gets instructions to execute the appropriate routines.

If KEYBOARD_IO finds a scan code representing an ASCII character (any ASCII character, including those between 128 and 255), it passes the ASCII number to a certain register in the 8088 and the scan code to another register. Then it tells the processor to resume running the program it was running when the interrupt occurred. That program may then avail itself of the character you've just sent.

If KEYBOARD_IO finds one of the special characters listed in Appendix G, it sends a 0 to the register that normally holds ASCII values and the scan code to the other register. When the current program regains control of the 8088 and finds a 0 in that first register, it knows to look at the second register and get your special character.

If scan codes arriving at KEYBOARD_IO represent neither an ASCII character nor a special character (nor one of the four characters that generate their own interrupts), then the ROM BIOS simply ignores your input and returns control to the current program.

There's more to say about the cooperative activities of the keyboard and the ROM BIOS, and much more to say about interrupts. To be continued in September. ▲

softalk presents the bestsellers

Now that it's all over except the shouting in the spreadsheet competition, it appears that the word processor arena is heating up once more.

The word crunching area has been hotly competitive all along, but the arrival of additional viable competitors from some of the strongest companies in the microcomputer industry raises the possibility of upsetting the status quo, which finds *WordStar* all alone, with everything trailing behind.

But first, a word about spreadsheets.

It now seems the veritable soul of conservatism to declare *1-2-3* the winner and new champion. What residual doubt existed because of *Multiplan*'s steady inroads into the market were extinguished in June. *1-2-3* widened its lead over Microsoft's contender, seemingly making it official.

Those who would say that placing *1-2-3* in the same category as your garden-variety spreadsheet is analogous to pitting Muhammad Ali against Sugar Ray Leonard should note that the only market equilibrium upset by *1-2-3*'s introduction was that of spreadsheets. Database products and word processors survived relatively unscathed and there were no graphing packages to speak of. Lotus's flagship product may be an integrated package in the sense that if it looks like a cow, gives milk, and subsequently tastes like beef, it must be a cow. But the market buys it as a spreadsheet, just as the food buyer purchases sirloin, irrespective of the fertilizer-spreading and milk-giving properties of the original animal.

Multiplan is a strong second, while *VisiCalc* awaits an update and *VisiOn* before making a move. The new, improved *SuperCalc* is managing to stem its losses, but gains are hard to come by. For the first month since *1-2-3* blasted the market, *MBA* lost significant ground.

But even as there's a decision in the spreadsheet market, uncertainty as to the status of the word processing genre increases. To be sure, *WordStar* remains dominant. But the array of challengers is growing.

Two questions stand out: Can *WordStar* hold on? Can the smaller companies compete on this battleground?

June sales saw the first positive results from *VisiWord* and *PFS:Write*, while *Word Perfect* and *PeachText* made significant inroads, sitting as the thirty-first and thirty-second programs in sales for June. How much room is there in a market that seemed already satisfactorily served?

Glut may not have described the pre-June situation, but there were *WordStar*, *EasyWriter 1.1*, *EasyWriter II*, *Volkswriter*, and *Word Plus-PC*, all doing well. *WordStar* has dominated just about every market it's ever served. The *EasyWriter* family is from Information Unlimited, a pioneer company in this area that's always been strong and competitive. *Volkswriter* and *Word Plus-PC* represent the sole significant offerings from their companies.

Into that fray enters VisiCorp with *VisiWord* and Software Publishing Corporation with *PFS: Write*. They'd be pretty hefty competition in any market, except possibly that of arcade games. *VisiWord* got out of the starting blocks fast, making sixteenth on the charts and being the highest-ranking newcomer to the Top Thirty. Even for VisiCorp, accustomed to success, that's pretty heady progress in such a competitive environment. *PFS:Write* was a much more modest starter, much like its older sibling *PFS:File* was in its early days. But like *File*, *Write* is getting good dealer reaction, which bodes well for its future.

Even as VisiCorp has previously dominated its spreadsheet markets, Peachtree has been the name in accounting packages for CP/M. Some time ago they bought rights to a highly regarded product called *Magic Wand* and transformed it into *PeachText*. It seems

to be growing on pc owners, and there's no doubt that Peachtree is a company with the will and experience to succeed.

There is more doubt about Perfect Software than there is about *Perfect Writer*. The product is getting high praise wherever it appears, but that's not in enough places. Perfect Software has gained more of a name for itself by tying its software to hardware. Can the company sell software when it isn't coattailing behind a hardware manufacturer? The consensus is that it can sure make software. It's the selling that remains to be seen.

But that's not all there is. Microsoft has its *Multi-Tool Word* ready to pounce on the market—mouse and all. *Multiplan*'s impacted traditional leaders in every market it's entered. Will *MTW* do unto *WordStar* what *Multiplan* did unto *VisiCalc*?

And you can't overlook another new product, *FriendlyWriter* from FriendlySoft. The Texans have scored twice in the entertainment market with product well designed for their intended audience. That's what product design and marketing are all about, and they may have something to say about the final outcome.

On another plane, MicroPro and IUS are accustomed to playing for keeps with the big guys. They'll battle VisiCorp, Microsoft, Peachtree, and Software Publishing down to the last pc owner. But what of Lifetree, Professional Software, Perfect Software and FriendlyWare? Are they prepared for the brouhaha to come?

The Instructor is becoming a dealer favorite. Many dealers are pressing this beginner's package on nearly every customer. They find it saves them significant after-sale service time. The program hit the top ten as a result of that support. ▲

IBM-franchised retail stores representing approximately 5.61 percent of all sales of IBM and IBM-related products volunteered to participate in the poll.

Respondents were contacted early in July to ascertain their sales for the month of June.

The only criterion for inclusion on the list was the number of units sold; such other criteria as quality of product, profitability to the computer store, and personal preference of the individual respondents were not considered.

Respondents in July represented every geographical area of the continental United States.

Results of the responses were tabulated using a formula that resulted in the index number to the left of the program name in the Top Thirty listing. The index number is an arbitrary measure of relative strength of the programs listed. Index numbers are correlative only to the month in which they are printed; readers cannot assume that an index rating of 50 in one month represents equivalent sales to an index number of 50 in another month.

Probability of statistical error is plus or minus 4.54 percent, which translates roughly into the theoretical possibility of a change of 4.83 points, plus or minus, in any index number.

the top thirty

This Month	Last Month	Index	
1.	1.	274.28	**1-2-3,** Mitch Kapor and Jonathan Sachs; Lotus Development
2.	2.	108.74	**Microsoft Flight Simulator,** Bruce Artwick; Microsoft
3.	3.	98.56	**WordStar;** MicroPro
4.	5.	54.10	**dBase II,** Wayne Ratcliff; Ashton-Tate
5.	4.	53.57	**Multiplan,** Microsoft; IBM
6.	7.	42.32	**PFS: File,** John Page and D. D. Roberts; Software Publishing Corporation
7.	8.	36.42	**Home Accountant Plus,** Mike Farmer, Bob Schoenburg, Larry Grodin, and Steve Pollack; Continental Software
8.	6.	33.21	**VisiCalc,** Software Arts/Dan Bricklin and Robert Frankston; VisiCorp; IBM
9.	20.	32.67	**The Instructor,** Jo-L Hendrickson; Individual Software
10.	10.	29.99	**EasyWriter II,** Basic Software Group; Information Unlimited Software
	15.	29.99	**Asynchronous Communications Support 2.0;** IBM
12.	9.	26.24	**Volkswriter,** Camilo Wilson; Lifetree
	21.	26.24	**Crosstalk;** Microstuf
14.	17.	25.17	**PFS: Report,** John Page; Software Publishing Corporation
	11.	25.17	**FriendlyWare P.C. Arcade,** Michael D. Yaw, James J. Davis, Alan Vanchura, Jr., R. B. Roberts, and Anthony Chumak; FriendlySoft
16.	—	23.57	**VisiWord;** VisiCorp
	15.	23.57	**Typing Tutor,** Michael Sierchio (Dick Ainsworth and Al Baker); IBM (Microsoft)
18.	13.	20.35	**Frogger,** Olaf Lubeck; Sierra On-Line
19.	—	18.21	**Smartcom II,** Hayes Microcomputer Products
	30.	18.21	**Personal Editor;** IBM
21.	18.	17.67	**Word Plus-PC,** Andres Escalon; Professional Software
22.	27.	17.14	**InfoStar;** MicroPro
23.	24.	15.53	**Cdex Training for VisiCalc,** Dr. Stephen C. Brandt; Cdex
24.	27.	14.46	**Norton Utilities,** Peter Norton; Peter Norton Inc.
	14.	14.46	**FriendlyWare P.C. Introductory Set,** Michael D. Yaw, James J. Davis, Frank Smith, Alan Vanchura, John Leatherwood, and Bruce W. Moore; FriendlySoft
26.	21.	11.24	**SuperCalc,** Sorcim
	—	11.24	**Zork I,** Infocom
	—	11.24	**KinderComp,** Doug Davis; Spinnaker Software
29.	—	10.71	**Basic Compiler,** Microsoft; IBM
30.	21.	10.17	**General Accounting,** John Moss and Ken Debower; IBM (BPI)

You Wouldn't Go to Work Without Your Pants

Why Buy an IBM-PC Without FAST FACTS

Your IBM-PC is only half-ready for work if it doesn't have FAST FACTS, the executive's filing system. FAST FACTS, you see, makes full use of the IBM-PC's capability to handle information quickly and efficiently. No matter if you're a business executive or home executive, FAST FACTS can take care of your filing needs. And with its built-in *Report Writer,* FAST FACTS, makes it a snap to summarize selected portions of your files. What's more, FAST FACTS can share information with VISICALC®, 1-2-3™ and WordStar®. It's also priced to fit easily in your computer budget.

So don't purchase an IBM-PC that's only half-ready for work. Get FAST FACTS, the latest introduction of Innovative Software, developers of proven executive businessware for the IBM-PC. (Ask your dealer about T.I.M. III and FAST GRAPHS, two other fine products from Innovative Software.)

Innovative Software® developers of practical businessware for the executive.

9300 W. 110th St., Suite 380
Overland Park, KS 66210 USA
913/383-1089 • Telex: 209542

Innovative Software's products are available at Computerlands®, Sears Business Centers, The Xerox Stores, Computer Marts, Entre' Computer Centers℠, CompuShops® and other quality dealers.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 673 NO. HOLLYWOOD, CA

POSTAGE WILL BE PAID BY ADDRESSEE

softalk

for the IBM Personal Computer

P.O. Box 60
North Hollywood, California 91603

softalk

for the IBM Personal Computer

1st year free to IBM Personal Computer owners.

PC Serial # ______________________________

Name ______________________________

Address ______________________________

City/State ______________________ Zip __________

☐ I don't own an IBM pc, but I would still like to subscribe. Payment enclosed.

☐ Visa ☐ MasterCard ______________________________

Acct. # Exp. Date

Signature ______________________________

Regular Subscription Rate: $24 per year.

Please allow 6-8 weeks for delivery of the first issue.

WE MUST HAVE YOUR SERIAL NUMBER TO PROCESS FREE SUBSCRIPTIONS

A SOFTALK PUBLICATION